UNIVERSIDAD DE CÁDIZ

Departemento de Ingeniería de Sistemas y Automática, Tecnología Electrónica y Electrónica

TESIS DOCTORAL

UN PROCEDIMIENTO PARA LA VERIFICACIÓN Y CLASIFICACIÓN PRIORIZADA DE FPGAs DE BAJO COSTE ANTE APLICACIONES SECTORIALES ELECTRÓNICAS

Autora: Mª Ángeles Cifredo Chacón

Directores: Dr. D.Diego Gómez Vela

Dr. D.Juan Manuel Barrientos Villar

Marzo de 2010

A mis padres Aurelio y Alicia.

A mi marido Javi.

Resumen

Esta tesis propone un método para establecer un orden de idoneidad, en distintos mercados, de dispositivos programables FPGA basado en un análisis de prestaciones frente a precio. Para determinar este orden se ha utilizado un conjunto de circuitos pertenecientes al repositorio público Opencores, descartando aquellos que han supuesto diferentes interpretaciones por parte de las herramientas de diseño. Como trabajo experimental, el análisis se ha extendido al subconjunto de modelos de FPGAs consideradas como "de bajo coste" de los fabricantes Xilinx y Altera, habiéndose empleado los entornos de desarrollo gratuitos ofrecidos por dichos fabricantes.

Para realizar el análisis se han desarrollado varias aplicaciones informáticas que permiten implementar y analizar de forma automática todos los circuitos sobre todos los dispositivos, leyendo los parámetros evaluables y filtrando las FPGAs por precio y características constructivas. Estas aplicaciones devuelven finalmente una clasificación basada en una puntuación que relaciona cinco magnitudes con el precio por unidad de cada dispositivo analizado.

El conjunto de circuitos utilizados como patrón es ampliable, así como el total de modelos de FPGAs tomados como muestra de estudio. Las aplicaciones desarrolladas aceptan estas modificaciones iniciando un nuevo proceso de puntuación y clasificación.

Abstract

This thesis proposes a method to establish a suitability order in different markets, for FPGA programmable devices based on a performance versus price analysis. A set of circuits belonging to Opencores public repository has been used to determine this order, discarding those that have been interpreted in different ways by design tools. As experimental work, the analysis has been extended to subset of FPGA devices considered as "low cost" from manufacturers Xilinx and Altera, having used the free development environments offered by these manufacturers.

Several software applications has been developed to perform the analysis, allowing the automatical implementation and analysis of every circuit on each device, reading the measurable parameters and filtering FPGA devices by price and design features. Finally, these applications return a classification based on a score that relates five quantities with the price per unit of each tested device.

The circuits used in the analysis and target FPGA devices are extensible. Developed applications accept these changes initiating a new score and ranking process.

Agradecimientos

Ante todo expresar mi agradecimiento a mis directores de tesis, Diego Gómez Vela y Juan Manuel Barrientos Villar. Su orientación, apoyo, consejo y experiencia han sido los pilares que sustentan este trabajo. Agradecer también su colaboración como compañeros del día a día en el trabajo docente e investigador que realizamos.

Al resto del Grupo de Diseño de Circuitos Microelectrónicos, en especial a Angel Quirós Olozábal que desinteresadamente me ha ayudado a solucionar muchos de los problemas con los que me he encontrado. A José María Guerrero, con quien he compartido durante muchos años el camino hasta finalizar nuestras tesis doctorales. A Luis Rubio Peña y a Francisco José Lucas Fernandez por crear un ambiente en el que ha sido fácil trabajar.

A todos los colaboradores del consorcio Opencores, cuyo trabajo ha permitido desarrollar esta tesis.

A Alberto Ramos Tijeras por su asesoramiento y colaboración en el tema financiero y bursátil, me hubiera gustado reflejar en esta tesis todas las ideas y conocimientos que me transmitió.

Finalmente a Javier Gómez de Cos, agradecerle tanto su experto consejo en los distintos problemas informáticos que me he encontrado, como su comprensión, apoyo moral y paciencia demostrada en mis malos momentos. Todo ello ha dado su fruto, por fin he terminado y podemos volver a compartir nuestro tiempo.

INDICE DE FIGURAS

1. INTRODUCIÓN

Los dispositivos lógicos programables denominados *Field Programmable Gate Array*, en adelante FPGAs, surgieron para llenar el hueco existente entre el diseño discreto y los circuitos integrados de aplicación específica. Se fabrican como componentes estándar que más tarde el diseñador personalizará para una determinada aplicación. Pueden ser programados tantas veces como sea necesario e incluso en el sistema final (*In-System Programability*), lo que las hace ideales para el desarrollo de prototipos de productos de cualquier mercado. Ofrecen ciclos cortos de desarrollo y por tanto reducen los tiempos de llegada al mercado sin renunciar a una alta fiabilidad. Sus precios y el consumo de potencia han disminuido y su rendimiento y prestaciones han aumentado. Ahora las FPGAs denominadas de bajo coste son capaces de albergar diseños complejos a un módico precio presentándose como una seria competencia para los ASIC.

Al principio, solo los productos con precios finales en torno a los miles de euros podían contar con FPGAs como elemento principal, quedando relegadas a servir solo como plataforma de prototipos. La fabricación de gran volumen tenía que recurrir a los ASIC debido principalmente a su ajustado coste unitario que permitía precios finales competitivos. Las modernas FPGAs con altas prestaciones, bajo consumo y un precio cada vez menor, son ideales para los mercados más sensibles al coste por unidad. Eliminan la incertidumbre que rodea a la fabricación de un ASIC cuyo ciclo de fabricación es extremadamente largo y durante el cual puede correr el riesgo de quedar obsoleto frente a las

necesidades del mercado, como consecuencia las ventas del producto no serían las esperadas(1). Con las FPGA de bajo coste, el producto final puede sufrir modificaciones para adecuarse a los nuevos requisitos sin que el precio final aumente Figura 1.

Tiempo de salida al mercado	Potencial de ventas
Primero en llegar	100%
3 meses tarde	73%
6 meses tarde	53%
9 meses tarde	32%
12 meses tarde	9%

Figura 1.Consideraciones sobre el tiempo de llegada al mercado

Dada las características de las FPGA son cada vez más los mercados que cuentan con ellas de algún modo en sus diseños. Edwards&Sons preguntó entre algunos ingenieros en qué porcentaje empleaban FPGAs en sus proyectos,repondiendo éstos que las incluían en un 81% de sus proyectos, los CPLD (*Complex Programmable Logic Device*) en un 69%, los ASIC estructurados en un 28% y en un 27% ASIC basados en celdas(2). In-Stat evaluó las ventas de FPGAs durante 2005, obteniendo una cifra de 1.900 millones de dólaresy una previsión de 2.750 millones de dólares para el presente año 2010.(3)

Entre los sectores usuarios de las FPGAs encontramos algunos extremadamente especializados como la industria aeronáutica y aeroespacial o la militar. Los fabricantes de FPGAs ofrecen productos exclusivos pensados para los exigentes entornos de trabajo de sus equipos como las altas temperaturas, la radiación o terrenos de gran dureza física. Otro de los grupos que tradicionalmente las utilizan en sus diseños es el de las comunicaciones, que casi siempre las ha incluido como lógica de adaptación.

La electrónica de consumo y la automoción son los mercados recientemente incorporados al mundo de las FPGAs. Son los grandes beneficiados de sus ventajas así como de la importante reducción de sus precios. Los cambios en los estándares de estos sectores son casi continuos y los dispositivos programables pueden actualizarse incluso alojados en el producto final. Esta flexibilidad representa una amenaza para los ASIC a pesar de sus grandes ventajas en costes, tamaño, fiabilidad y rendimiento.

Las FPGAs se han ido adaptando a los requisitos de la automoción como son las temperaturas extremas o el cumplimiento de algunas normativas como la AEC-Q100 (4) y ofrecen series especiales para este sector cada vez más interesado en ellas.

Otra causa del aumento de ventas de las FPGAs es el despegue tecnológico de algunos países asiáticos como India y China. Las comunicaciones tanto inalámbricas (instalaciones WiFi ó Wimax) como de cable (DSL o fibra hasta el hogar), la telefonía móvil y la televisión de alta definición, han potenciado enormemente el uso de estos dispositivos en este continente.

Los fabricantes de FPGAs diseñan sus productos pero la fabricación la contratan en otras empresas especializadas en manufactura de circuitos integrados. A lo largo de la historia han existido hasta cuarenta y tres empresas dedicadas a los dispositivos programables, la mayor parte de ellas ya no existen o han dejando sus divisiones de productos programables, Figura 2. A día de hoy sobreviven como fabricantes de FPGAs las siguientes, citadas por orden alfabético:

- Actel Corp.
- Altera Corp.
- Lattice Semiconductor
- Xilinx, Inc.

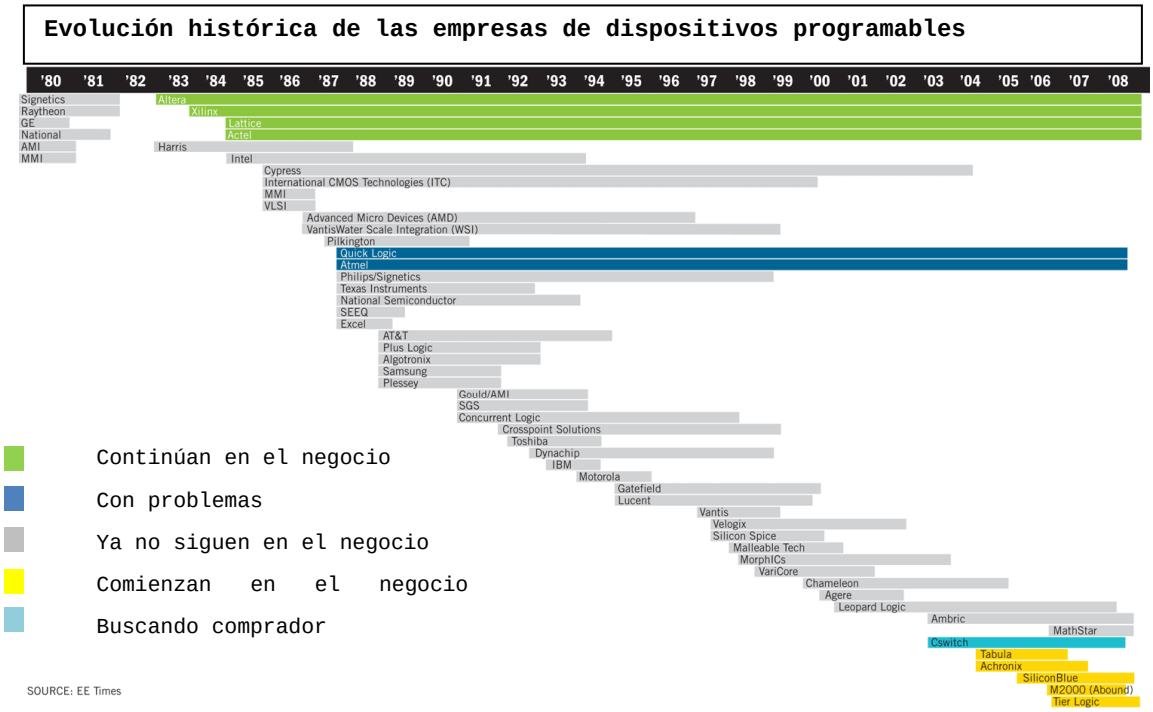


Figura 2.Trayectoria temporal de fabricantes de FPGA (Fuente: EETimes)

Recientemente han aparecido nuevas empresas en el sector como Silicon Blue o Achronix. De la lista anterior, Altera y Xilinx ocupan un 86,7% de las ventas mundiales y las demás compiten por el 13,3% restante (5).

Al principio estas dos empresas obtenían la mayor parte de los beneficios gracias a sus FPGAs de alto rendimiento y no a las de bajo coste muy limitadas entonces. No les interesaba que sus series económicas incluyeran características que pudieran hacer sombra a sus grandes protagonistas. Esta circunstancia fue aprovechada por el resto de empresas que no tenían nada que perder y apostaron por servir FPGAs de bajo coste con grandes prestaciones. Introdujeron en el mercado FPGAs económicas con mayor densidad, memoria no volátil en el chip, bloques DSP, o

pines de entrada/salida de alta velocidad, entre otros. Algunos sectores, lo más sensibles a costes, como el de la automoción o la electrónica de consumo, acogieron con agrado estos nuevos productos. Esta circunstancia obligó a que los proveedores dominantes retomaran el interés por sus FPGAs de menor precio.

A día de hoy, la variedad de FPGAs es enorme en cuanto a arquitectura se refiere e incluyen entre sus recursos, además de circuitos combinacionales y secuenciales, elementos tales como los que se citan a continuación y que pueden verse en la Figura 3:

- Procesadores (no disponibles en FPGAs de bajo coste)
- Procesadores digitales de señal (DSP)
- Multiplicadores
- Drivers de Entrada/Salida de alta velocidad
- Bloques de Memoria
- Bloques de gestión de reloj (PLLs o DLLs)

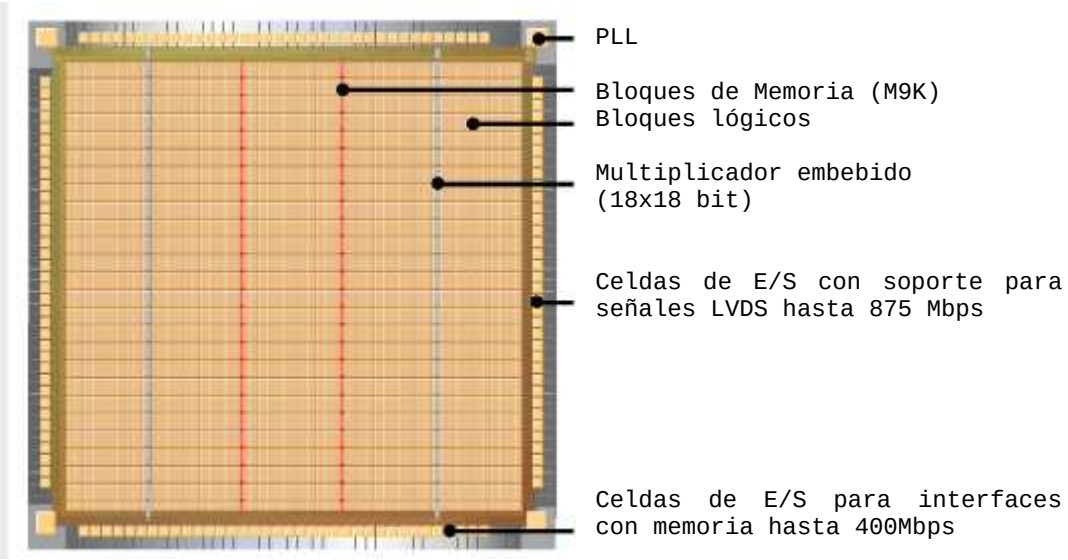


Figura 3. Distribución espacial de los recursos de la FPGA Cyclone III (Fuente: www.altera.com)

Además del gran número de recursos hardware con que cuentan las FPGAs, existen componentes virtuales denominados *Intelectual Property*, en adelante IP, que de forma simple pueden agregarse al diseño y sintetizarse sobre una FPGAs. Los módulos IP son descripciones software realizadas en lenguajes HDL (*Hardware Description Language*) que evitan al usuario tener que realizar los bloques más comunes de su diseño así como comprobar su correcto funcionamiento. Son reutilizables en posteriores proyectos lo que rentabiliza la inversión en ellos. En los últimos años, la demanda de estos núcleos virtuales ha crecido y han surgido empresas dedicadas exclusivamente a la comercialización de los mismos. Algunas de ellas actúan de forma autónoma como CAST Inc. y otras se asocian con los principales fabricantes de FPGAs como *Helion Technology Limited* asociada con *Xilinx INC*. ó *Arrow Electronics* que colabora con Altera.

Por otro lado, Internet ha propiciado la existencia de módulos IP gratuitos gracias a grupos impregnados por la filosofía del software/hardware abierto. Una muestra de ello es el consorcio conocido como OpenCores fundado en 1999 por un grupo de ingenieros y estudiantes liderados por Damian Lampret que desde entonces aportan y distribuyen núcleos IP de diversas funcionalidades.

Las herramientas de diseño junto con las tarjetas de desarrollo han jugado también un papel importante en la promoción de las FPGAs. El fabricante las adapta a sus tecnologías, las hace fáciles de manejar e incluso consiguen que aquellos que solían trabajar con ASICs apenas noten el cambio. Sus ofertas incluyen versiones reducidas gratuitas que en ocasiones pueden ser suficientes para ciertos diseños, y en otros casos no al carecer de soporte para algunas familias tecnológicas. Xilinx ofrece el entorno de desarrollo ISE Design Suite 11 y como versión reducida gratuita WebPack ISE 10. Altera denomina a su herramienta de diseño Quartus II v.9.0 y ofrece el paquete gratuito Quartus II web edition. Actel distribuye la aplicación

Libero IDE 8.4 para la que ofrece una licencia gratuita aunque limitada a un tiempo máximo de un año y LatticeSemiconductor denomina a su herramienta gratuita ispLever Starter, aunque solo es válida durante seis meses.

Ante tantas opciones de modelos y fabricantes, la elección de un modelo de FPGA que se ajuste a las necesidades del diseño a un precio asequible, requiere un estudio exhaustivo. A continuación se citan algunas de los criterios que se suelen barajar:

Parámetros	Soluciones
Temperatura, radiaciones, medio físico hostil	Series especiales
Potencia	Dispositivos de bajo consumo, especialmente para dispositivos portátiles
Tamaño del diseño	Influye en la densidad lógica y el número de terminales
Tamaño del sistema final	Solución en un chip
Actualización	Tecnologías reprogramables: SRAM, FLASH
Diseño de alto rendimiento ó mínimo precio por unidad	Series de alto rendimiento o de bajo coste
Elementos de seguridad	Encriptación, código de seguridad, etc.

Tabla 1. Consideraciones sobre los dispositivos programables

Los productos tecnológicos actuales tienen tiempo de vida limitado y es necesario reemplazarlos o modificarlos rápidamente. Esta circunstancia obliga a las empresas del sector a conseguir ciclos de diseño y fabricación reducidos. El tiempo se convierte en una variable crítica en el éxito o el fracaso del producto. Como ejemplo el sector de la telefonía móvil, cuyos productos están bien establecidos en el mercado pero que requieren de continuas renovaciones para garantizar las ventas.

Si una empresa decide incluir FPGAs en su diseño debe evaluar todos los modelos existentes y decidirse por aquella tecnología que le asegure el menor precio, la mayor calidad, el mejor rendimiento y en muchos casos el menor consumo. Estudiará los parámetros anteriormente citados e intentará comprar aquel modelo que aloje adecuadamente su diseño. El tiempo empleado en esta tarea se sumará al ciclo de diseño y fabricación pudiendo ser suficiente para que el producto no sea pionero, o bien que el nuevo modelo de un producto existente llegue tarde al punto de venta habiendo optado el comprador ya por otra marca.

Para ayudar a la elección entre diferentes modelos de un mismo producto se crearon los patrones de comparación (en adelante PdC). Elementos tales como procesadores de propósito general, procesadores embebidos o de procesamiento digital de la señal hacen uso de ellos. Estos patrones son también conocidos como *benchmarks* y son propuestos normalmente por organizaciones creadas al efecto. Los PdC someten a un elemento o sistema a una serie de pruebas para medir su rendimiento según un determinado parámetro. Uno de los más populares es el denominado SPEC (*Standard Performance Evaluation Corporation*) que puntúa el rendimiento de un procesador tras compararlo con otro de referencia. Esta puntuación facilita al comprador la decisión de qué artículo adquirir.

La elaboración de un PdC fiable para FPGAs que facilite la elección de uno u otro modelo, ha sido una idea que los fabricantes han mantenido en mente desde que apareciera en el mercado la primera FPGA en el año 1985. Dada la dificultad de esta empresa, no han sido demasiadas las propuestas de PdCs publicados, algunos de ellos se refieren en el capítulo siguiente.

1.1. Objetivos de la tesis

La transición ocurrida en los últimos cinco años en el campo del diseño microelectrónico, ha conducido a la preponderancia del empleo de componentes programables del tipo FPGA, que se han establecido en sectores de aplicación como Comunicaciones, Automoción, Juegos y Electrónica de consumo.

Esta evolución hace que el interés del diseñador sea resolver una aplicación en el menor tiempo posible y al menor coste, inhibiéndose de la problemática de la verificación del chip que acogerá su diseño, pues esto ya lo habrá hecho el fabricante del mismo con anterioridad. Liberado de esa tarea el diseñador se encuentra con una pléyade de componentes aptos o no para su propósito y una dinámica de competitividad en cuanto a coste, generada en la pugna comercial de las tres o cuatro firmas líderes del sector, que ha de conocer, actualizar y tomar una decisión en el menor tiempo posible.

La dinámica actual donde la obsolescencia de los productos y la presión del mercado obligan a un rediseño frecuente, o a una actualización para incrementar prestaciones, en tiempos de desarrollo cada vez más cortos, plantea la necesidad de disponer de una herramienta que explore el espacio de diseño desde el punto de vista costo-prestaciones. Con ese propósito disponer de una clasificación de acuerdo a una función de coste es un objetivo del mayor interés y su logro una aportación científica nada trivial.

Las nuevas técnicas de diseño a nivel de sistema (*ESL*, *Electronic System Level*), están permitiendo que las ventajas inherentes al diseño con FPGAs dejen de ser patrimonio exclusivo de los ingenieros de hardware y pasen a ser una alternativa viable para los ingenieros de software que consiguen mejoras en el rendimiento de sus algoritmos gracias a los beneficios del

paralelismo del hardware. Especialistas ajenos a los lenguajes HDL consiguen implementar sus diseños en dispositivos programables mediante los nuevos lenguajes de alto nivel (HLL) para síntesis, todo consumiendo un tiempo muy pequeño comparado con las técnicas más antiguas e incluso posibilitando la existencia de una solución hardware-software monochip (*System on a chip, SoC*). Disponer de una clasificación de FPGAs simplificaría aún más la difícil tarea de colocar un producto en el mercado justo a tiempo mediante la técnica ESL (6).

El presente trabajo pretende hacer una aportación a la difícil tarea de elegir un modelo de FPGA proponiendo una metodología que permita disponer de circuitos públicos que actúen como patrón puntuándose el rendimiento de los dispositivos ensayados mediante una métrica que relacione los habituales parámetros de velocidad, área y potencia con un cuarto no menos importante como es el precio del elemento. Los objetivos perseguidos son:

a) Identificar los principales sectores comerciales que hacen un mayor uso de electrónica programable del tipo FPGA.

b) Concretar qué tipos de dispositivos programables FPGA presentaron una mayor demanda en los últimos años para incluirlas o no como tecnologías objetivo en los test a realizar.

c) Determinar un conjunto de circuitos de dominio público descritos imparcialmente para evitar polarizar los resultados y que sean interpretados en igualdad de condiciones por las herramientas de diseño. Clasificar estos circuitos según su demanda en los mercados elegidos.

d) Evaluar las herramientas existentes para determinar el método de trabajo con las mismas de modo que devuelvan resultados equiparables. Valorar la idoneidad para el presente trabajo de utilizar herramientas gratuitas así como comprobar la bondad de los resultados conseguidos empleando las distintas estrategias propuestas por los propios fabricantes.

e) Establecer cuáles serán los parámetros asociados al rendimiento de las tecnologías ensayadas y el método para conseguir una lectura justa de los mismos en las herramientas empleadas. Estudiar las interrelaciones que pudieran existir entre los parámetros para conformar una función de coste en base a la cual puntuar y clasificar todas las tecnologías probadas.

f) Automatizar las tareas que a continuación se citan, dada la inexistencia de herramientas específicas para ello y el carácter repetitivo de los ensayos, asi como la necesidad de asegurar resultados fiables.

	Tareas	Ejemplo
1	Ejecución automática de las fases de síntesis e implementación, así como de los análisis de tiempo y potencia	• Circuitos del PdC (C_i): C1, C2, …Cn • Tecnologías (F_j) : F1, F2,..F_m • Síntesis, implementación y análisis = n * m • Informes de interés (I_k)=I_1,I_2, … I_p • Datos representativos(D_l): D_1,D_2, … ,D_q Supuesto 10 circuitos, 100 tecnologías,5 informes, 2 datos por informe se generan:
2	Lectura de una serie de parámetros en los informes generados por las herramientas tras concluir la tarea anterior	
3	Tratamiento de datos y homogeneización de los formatos entre los resultados de los distintos fabricantes	
4	Cálculo de la puntuación conseguida por cada tecnología ante los circuitos del PdC	

	Tareas	Ejemplo
5	Asistente de ayuda a la selección de FPGAs basado en criterios constructivos y en la puntuación conseguida con el PdC.	- 10*100 = 1.000 Ejecuciones de las herramientas - 1000*5 = 5.000 informes generados - 5000 * 2 = 10.000 lecturas de datos A continuación comenzaría el proceso de cálculo de puntuaciones en base a la información obtenida. La automatización optimiza el trabajo y evita posible errores.

Tabla 2. Descripción de las tareas susceptibles de automatización

1.2. Organización de la tesis

El **capítulo 2** describe las propuestas que han existido o existen para realizar comparativas de procesadores de propósito general, embebidos y de procesamiento digital. De forma más detallada se expondrán las iniciativas que a lo largo de la historia han surgido para evaluar y comparar las distintas FPGAs, así como todas las dificultades que supone esta complicada tarea.

El **capítulo 3** ofrece una introducción a las FPGAs de bajo coste repasando el inicio de su éxito en el mercado, la importancia y trayectoria de sus fabricantes, sus aplicaciones clásicas y las más novedosas y resumirá las principales diferencias entre las arquitecturas existentes. Por último presentará el subconjunto de FPGAs tomado como muestra para este trabajo.

En el **capítulo 4** se expone la metodología de selección de circuitos IP, siguiendo estos pasos se construirá, para esta ocasión, un PdC concreto que actuará como objetivo para ensayar sobre el grupo de FPGAs obtenido. A continuación se explica el procedimiento a seguir para ejecutar el PdC: pasos previos, nombre y sintaxis de los ficheros de automatización, comandos y configuración que incluyen según herramienta. La implementación trata de conseguir el mejor rendimiento posible de cada dispositivo con cada circuito pero manteniendo la premisa de una mínima intervención del usuario, de tiempo y de dinero.

El **capítulo 5** reflexiona sobre las magnitudes relevantes en los dispositivos programables FPGA según la perspectiva planteada desde el inicio de este trabajo. En base a ellas se formulan propuestas para puntuar las FPGAs por cada magnitud en relación a su precio, asi como la expresión que evalúe el rendimiento total de cada modelo ante el patrón de comparación establecido.

En el **capítulo 6** se incluye toda la información relativa a las aplicaciones informáticas desarrolladas para leer los resultados, calcular las puntuaciones y presentar resultados basados en criterios de búsquedas proporcionado por el usuario.

El **capitulo 7** recoge los resultados experimentales obtenidos tras la ejecución del PdC, leyendo todas las magnitudes relevantes y en base a ellas efectuando los cálculos de puntuaciones oportunos con la ayuda de las aplicaciones desarrolladas.

Finalmente el **capitulo 8** recopila las conclusiones obtenidas basadas en los resultados expuestos en el capitulo anterior, concluyendo sobre su viabilidad, fiabilidad e interés. Revisa la consecución de los objetivos iniciales e incluye nuevas propuestas para el futuro.

2. ANTECEDENTES

2.1. Definición de patrón de comparación

La definición genérica de este término es “Patrón, punto de referencia o criterio respecto al cual la calidad de algo puede ser medida, juzgada y evaluada”

La palabra *benchmark* es la palabra que habitualmente se emplea en los países angloparlantes y puede ser traducido al castellano como comparativa. Proviene de las palabras *bench* (banquillo, mesa) y *mark* (marca, señal). El uso del término tiene su origen en la Inglaterra del siglo XIX, cuando los agrimensores hacían una hendidura en una piedra de un muro y dentro de la misma introducían un ángulo metálico que formaba algo similar a un banco. Sobre él colocaban una vara para medir la altura o nivel de ese terreno, de esta forma podían volver a colocar la vara exactamente en el mismo lugar. La marca quedaba señalada por unas flechas talladas en la piedra bajo el corte hecho, tal y como se aprecia en la Figura 4.

Figura 4.Marca de los agrimensores ingleses del siglo XIX (Ordenance Survey)

Ya en entornos más técnicos, se denomina PdC al conjunto de pruebas a que se somete un elemento o sistema para medir su rendimiento según un determinado parámetro. El campo de los procesadores informáticos ha sido donde esta técnica de comparación ha tenido un mayor desarrollo. En este sector, un PdC es el resultado de la ejecución de un programa o un conjunto de ellos en una máquina, con el objetivo de estimar el rendimiento de un elemento concreto o la totalidad de la misma, y así poder comparar los resultados con máquinas similares. En términos de ordenadores, un PdC podría ser realizado en cualquiera de sus componentes, ya sea CPU, RAM, tarjeta gráfica, etc. También puede ser dirigido específicamente a una función dentro de un componente, por ejemplo, la unidad de coma flotante de la CPU o incluso a otros programas como un compilador.

La técnica de utilización de PdC poco a poco, ha ido extendiéndose hacia otros dispositivos electrónicos como son los procesadores digitales de señal, microcontroladores empotrados en dispositivos programables, circuitos integrados, sistemas en un chip, sistemas de señal mixta (7) o sistemas de reconocimiento de voz entre otros. Tímidamente han aparecido algunos PdC para dispositivos lógicos programables, más adelante se revisan las propuestas que han existido con el objetivo de comparar FPGAs.

2.2. Patrones de comparación en circuitos integrados

2.2.1. ISCAS'85 e ISCAS'89

Otro escenario donde ha habido propuestas de PdC es el de los circuitos integrados, donde ISCAS'85 e ISCAS'89 son los más

populares. El primero de ellos fue presentado por Brglez y Fujiwara y consistía en un conjunto de diez circuitos digitales representados a nivel de puertas (7). La segunda versión fue ofrecida por Brglez, Bryan y Kozminski que ampliaron el juego a treinta y un circuitos (8). Dado el carácter público de su descripción a nivel de puertas, han sido utilizados para investigar sobre problemas como la generación automática de test (ATPG), simulación de fallos y compresión de datos.

2.2.2. ITC'99

Iniciativa que intentó crear un nuevo conjunto de circuitos para el test de circuitos integrados sustituyendo a ISCAS'85 e ISCAS'89.Se componía de un total de treinta circuitos procedentes de instituciones académicas y empresas. Los circuitos eran tanto combinacionales como secuenciales y estaban descritos a nivel de puertas y a nivel de registros. Algunos incluían memoria embebida y otros no, y solo algunos eran de libres disposición (9)(10).

2.2.3. ITC'02 SOC

La complejidad de los circuitos integrados sigue creciendo y a día de hoy es posible integrar en un único dado la funcionalidad completa de un determinado sistema. Se les denomina sistema en un chip, *system on chip*, y en adelante SOC. Pero esta tendencia supone un reto para el proceso de test de estos circuitos integrados debido a lo heterogéneo de sus componentes y al propio tamaño. El test modular aparece como una interesante opción frente al test de sistema, entre otras razones debido a la imposibilidad de acceder a la implementación de algunos de los núcleos integrantes del SOC.

ITC'02 está destinado a la investigación y comparación de rendimiento y efectividad de métodos y herramientas del test

modular de SOC. Trata de combinar la fortaleza de ISCAS y de ITC'99 ofreciendo un total de doce modelos de SOC con un número diferentes de núcleos ofrecidos por nueve importantes empresas como Philips, Texas Instruments o Hewlett Packard (11) entre otras.

2.3. Patrones de comparación de procesadores y sistemas basados en procesador

2.3.1. SPEC

SPEC es el acrónimo de *Standard Performance Evaluation Corporation*, es una organización sin ánimo de lucro fundada en 1988 por un pequeño número de vendedores de estaciones de trabajo que vieron la necesidad que tenía el mercado de contar con una serie de patrones de referencia para establecer comparativas. Su estructura cuenta con tres grupos:

- OSG (*Open System Group*): es el comité original, este grupo se encarga de los PdCs para servidores, estaciones de trabajo y equipos de sobremesa. Sus conjuntos de programas de pruebas son los más populares y reciben el nombre de SPEC CPU 2006.

- HPG (*High Performance Group*): este grupo se dedica a los PdC de sistemas de equipos de alto rendimiento: multiprocesadores simétricos, clusters de estaciones de trabajo, sistemas paralelos con memoria distribuida y los tradicionales supercomputadores vectoriales y vectoriales paralelos.

- GPC (*Graphics Performance Characterization*): caracterización del rendimiento de gráficos, realizan PdCs para subsistemas gráficos.

Actualmente cuenta con más de sesenta compañías entre sus integrantes como Advanced Micro Devices, Apple Computer Inc., Dell, Fujitsu Siemens, Hewlett-Packard, IBM, Intel o Microsoft entre otros. El principal objetivo de SPEC es asegurarse de que existan una serie de métricas justas y útiles que permitan diferenciar entre distintos sistemas, de modo que el comprador pueda establecer comparativas en función de sus necesidades. A su vez, permite a los fabricantes mejorar sus productos ofreciendo mejores rendimientos.

SPEC solo especifica los programas que se van a utilizar para medir prestaciones pero no los ejecuta, confía en los fabricantes para hacerlo, más tarde supervisan los resultados y los publican en su página de internet (12).

Los PdC de SPEC han pasado por distintas versiones, la última de las cuales es la SPEC2006, las anteriores fueron SPEC89, SPEC92, SPEC95 y SPEC2000; siempre hay que tener en cuenta la versión a la hora de acudir a este referente.

Dentro de SPEC2006 existen dos conjuntos principales de pruebas que son:

- **CINT2006** un total de doce programas extraídos de aplicaciones reales en lugar de usar núcleos de códigos sintéticos. Miden y comparan el rendimiento de computación intensiva con enteros. Nueve de ellos están escritos en C y el resto en C++. Cada componente del conjunto pertenece a distintas áreas como la programación, inteligencia artificial, juegos o compresión de vídeo entre otras.

- **CFP2006** conjunto de diecisiete programas para medir y comparar el rendimiento de computación intensiva de flotantes. Escritos en C, C++ y Fortran y extraídos de aplicaciones reales de carácter científico como pronóstico

del tiempo, estructuras mecánicas, dinámica de fluidos, etc.

La "C" en CINT2006 y CFP2006 denota que son PdCs a nivel de componentes, en oposición a PdCs de todo el sistema. Miden el rendimiento del procesador, del sistema de memoria y del compilador.

SPEC CPU2006 usa como referencia la máquina de 1997 "Ultra Enterprise 2" con un procesador de 296 MHz UltraSPARCII. Todos los códigos de pruebas son ejecutados en esta computadora y los tiempos obtenidos en ella son tomados como referencias del PdC. Cuando el conjunto de programas se ejecute en otro equipo, se le asignará una puntuación SPEC según la relación entre sus resultados y los obtenidos por Ultra Enterprise 2. Así cuanto mayor sea el valor asignado mejor será el resultado.

Las métricas que ofrece SPEC2006 son de velocidad y productividad. En la primera se compara la capacidad de un procesador para realizar tareas simples comprobando si completa todos los códigos que componen CINT2006 y CFP2006. Para estudiar la productividad del procesador se mide el número de tareas, CINT y CFP, que realiza en un determinado tiempo. Existen dos caminos para realizar ambas comparativas, una más estricta pero más simple para el usuario (base) y otra sin tantas restricciones (*peak*) que maximiza el rendimiento. Estos resultados son publicados una vez han sido ejecutados por los distintos fabricantes.

Los valores que ofrecen son *Seconds* y *Rate,* el primero informa del tiempo que llevó al equipo ejecutar ese código, el segundo la relación entre ese valor y el de la máquina de referencia. Cada PdC debe ejecutarse tres veces y aunque todos los resultados se publican se destaca el valor central de los tres obtenidos ya que más tarde se usará para calcular la puntuación SPEC. Para calcular la puntuación final se obtiene la media

geométrica de los ratios asociados a los tiempos intermedios de ejecución de cada programa (Xi):

$$\boldsymbol{Ratio\ (Xi) = (Tiempo\ de\ referencia / Tiempo\ de\ la\ máquina\ bajo\ test)} \quad \textbf{(Ecuación 1)}$$

$$\boldsymbol{Puntuación\ SPEC = \sqrt[n]{\prod_{i=1}^{n} Xi}} \quad \textbf{(Ecuación 2)}$$

SPEC® CINT2006 Result

Copyright © 2007 Standard Performance Evaluation Corporation

Fujitsu Siemens Computers
CELSIUS M460, Intel Core 2 Quad Q6700 processor

SPECint®2006 = 20.8
SPECint_base2006 = 18.6

CPU2006 license: 22
Test sponsor: Fujitsu Siemens Computers
Tested by: Fujitsu Siemens Computers

Test date: Dec-2007
Hardware Availability: Nov-2007
Software Availability: Nov-2007

Benchmark	Base						Peak					
	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio
400.perlbench	583	16.8	579	16.9	**580**	**16.8**	**451**	**21.7**	452	21.6	451	21.7
401.bzip2	**683**	**14.1**	696	13.9	681	14.2	651	14.8	**646**	**14.9**	644	15.0
403.gcc	**489**	**16.5**	489	16.5	483	16.7	430	18.7	416	19.4	**426**	**18.9**
429.mcf	387	23.6	**387**	**23.6**	387	23.5	387	23.6	**387**	**23.6**	387	23.5
445.gobmk	613	17.1	613	17.1	**613**	**17.1**	**532**	**19.7**	532	19.7	532	19.7
456.hmmer	**666**	**14.0**	666	14.0	666	14.0	470	19.9	470	19.8	**470**	**19.8**
458.sjeng	764	15.8	**765**	**15.8**	765	15.8	686	17.6	**686**	**17.6**	687	17.6
462.libquantum	443	46.8	**443**	**46.8**	443	46.8	443	46.8	**443**	**46.8**	443	46.8
464.h264ref	793	27.9	794	27.9	**794**	**27.9**	**744**	**29.7**	744	29.7	744	29.7
471.omnetpp	435	14.4	**435**	**14.4**	436	14.3	**390**	**16.0**	390	16.0	390	16.0
473.astar	532	13.2	**532**	**13.2**	532	13.2	**477**	**14.7**	477	14.7	476	14.7
483.xalancbmk	344	20.0	345	20.0	**344**	**20.0**	342	20.2	342	20.2	**342**	**20.2**

Figura 5. Resultados del PdC CINT2006 de velocidad

Los resultados de productividad se obtienen ejecutando un total de n copias del mismo programa concurrentemente, siendo n normalmente el número total de núcleos que posee el procesador en estudio.

Results Table														
Benchmark	Base							Peak						
	Copies	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Copies	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio
400.perlbench	4	**620**	**63.0**	618	63.2	621	62.9	4	478	81.7	**478**	**81.7**	480	81.5
401.bzip2	4	852	45.3	846	45.6	**851**	**45.4**	4	**820**	**47.1**	821	47.0	807	47.9
403.gcc	4	667	48.3	655	49.2	**666**	**48.3**	4	666	48.3	**676**	**47.6**	691	46.6
429.mcf	4	740	49.3	**740**	**49.3**	739	49.4	4	740	49.3	**740**	**49.3**	739	49.4
445.gobmk	4	609	68.9	609	68.9	**609**	**68.9**	4	**556**	**75.4**	558	75.2	556	75.5
456.hmmer	4	673	55.4	673	55.5	**673**	**55.4**	4	506	73.8	**511**	**73.0**	512	72.9
458.sjeng	4	796	60.8	**797**	**60.8**	797	60.8	4	713	67.9	717	67.5	**717**	**67.5**
462.libquantum	4	1733	47.8	**1732**	**47.9**	1732	47.9	4	1733	47.8	**1732**	**47.9**	1732	47.9
464.h264ref	4	**820**	**108**	821	108	819	108	4	776	114	787	112	**779**	**114**
471.omnetpp	4	796	31.4	796	31.4	**796**	**31.4**	4	762	32.8	766	32.7	**765**	**32.7**
473.astar	4	**726**	**38.7**	726	38.7	726	38.7	4	679	41.4	681	41.3	**680**	**41.3**
483.xalancbmk	4	455	60.6	455	60.7	**455**	**60.7**	4	444	62.1	446	61.8	**446**	**61.8**

Results appear in the order in which they were run. Bold underlined text indicates a median measurement.

Figura 6. Resultados del test CINT2006 de productividad para la misma máquina

2.3.2. Patrones de comparación para procesadores, microcontroladores y compiladores embebidos (EEMBC®)

El consorcio, sin ánimo de lucro, EEMBC® (*Embedded Microprocessor Benchmark Consortium*), fue fundado en 1997 por Markus Levy. La necesidad de un nuevo conjunto de patrones que estudiaran el rendimiento de los procesadores embebidos era cada vez mayor, ya que los antiguos Dhrystone MIPS no se ajustaban demasiado bien a estos nuevos circuitos. Entre sus miembros se encuentran empresas como AMD, ARM, Microchip, EDN, Hitachi, IBM, Intel y un largo etcétera hasta completar las cincuenta que actualmente forman el consorcio.

El conjunto de PdCs propuesto por EEMBC® se ha establecido como el estándar industrial para evaluar las capacidades de los procesadores embebidos, compiladores e implementaciones Java siguiendo criterios claros y reales. Pretenden orientar al diseñador a la hora de escoger el procesador embebido más adecuado para su sistema. Está formado por un conjunto de algoritmos y aplicaciones extraídas de situaciones del mundo real y orientado hacia distintos sectores como la automoción,

telecomunicaciones, electrónica de consumo, aplicaciones Java y las redes.

A diferencia de SPEC, EEMBC® vuelve a ejecutar el conjunto de PdC ya realizado por el fabricante que desea publicar los resultados. De este modo asegura la repetitividad de la puntuación, su fiabilidad y el cumplimiento de las reglas establecidas. Las puntuaciones obtenidas por los distintos procesadores puede encontrarse en la página web de EEMBC® (12).

Como ya se ha mencionado estos PdC están organizados por sectores industriales. En cada grupo se incluyen los algoritmos considerados más frecuentes en cada mercado. Existe un total de siete grupos que son:

- **Automoción**. AutoBench es un conjunto de PdCs que permite a los usuarios predecir el rendimiento de los microprocesadores y microcontroladores en aplicaciones industriales, de automoción y de propósito general. Incluye un total de dieciséis algoritmos sobre manipulación de bits, modulación de ancho de pulso (PWM), operaciones de multiplicación y de desplazamiento; otros más específicos de la automoción como la comunicación mediante bus CAN, o el cálculo de la velocidad en carretera, además de otros de procesamiento digital de la señal (FIR, FFT, DCT, IDCT, etc).

- **Electrónica de consumo**. ConsumerBench es el conjunto de PdC que permite evaluar los procesadores de cámaras digitales, impresoras u otros sistemas embebidos que manejan imágenes digitales. Incluye cuatro algoritmos dedicados a la compresión/decompresión de imágenes, según el estándar JPEG, así como algoritmos para la conversión y el filtrado de color.

- **Audio y video**. DENBench estudia el rendimiento de procesadores en aplicaciones que incluyen tratamiento de imagen, video y audio, encriptación y comercio electrónico. Orientados a productos como reproductores de MP3 o de DVD, telefonía móvil, PDAs, cámaras digitales, decodificadores de TV o sistemas de entretenimiento en vehículos. Incluye algoritmos relativos a cuatro subgrupos:

    - ✓ MPEG: decodificación MPEG2 y MPEG4

    - ✓ Criptografia: algoritmos de encriptación estándar como AES (*Advanced Encryption Standard*), DES (*Data Encryption Standard)* o Huffman entre otros.

    - ✓ Procesamiento digital de imagen: JPEG y conversión del espacio de color, RGB to YIQ, etc.

    - ✓ Codificación MPEG en coma flotante.

- **Aplicaciones Java para dispositivos portátiles.** GrinderBench (Java) es un conjunto de PdCs que permite al usuario evaluar el rendimiento de las aplicaciones basadas en Java 2 Micro Edition (J2ME) para productos tales como la telefonía móvil y las PDAs. Está formado por cinco grupos de algoritmos:

    - ✓ Algoritmos que representan jugadas de ajedrez estimando el tiempo empleado en ello.

    - ✓ Algoritmos que miden el rendimiento de las aplicaciones Java creadas para las transacciones encriptadas, como el pago por móvil.

    - ✓ Algoritmos de lectura y conversión de elementos XML para ser manejados por JavaScript.

- ✓ Tareas paralelas de interfaz con el usuario e interacción con internet

- ✓ Algoritmos que estudian la velocidad de decodificación de una imagen fotográfica de tamaño habitual en la telefonía móvil.

- **Redes**: Networking permite evaluar la velocidad a la que un procesador envía paquetes en aplicaciones de red. Siete grupos de algoritmos conforman este PdC con funciones tales como comprobación de la cabecera de las tramas, reagrupación de paquetes, gestión de envíos según reglas NAT, protocolo TCP y algoritmos que simulan la comprobación de calidad de servicio (QoS) orientado a la transmisión de multimedia, voz y video.

- **Sistemas de oficina**. OABench comprueba el rendimiento de los procesadores de impresoras, plotters u otros equipos dedicados al procesamiento de texto e imagen propios de una oficina. Entre los algoritmos que lo componen están los dedicados a evaluar el lenguaje de control, rotación de texto o impresión PostScript entre otros.

- **Telecomunicaciones.** Telebench evalúa procesadores de las aplicaciones de comunicación. Los algoritmos empleados son propios del campo DSP: autocorrelación, Viterbi, FFT, iFFT, corrección de errores para mejorar el rendimiento de las implementaciones propias de radio digital, bluetooth o telefonía móvil, o bien algoritmos de transmisión sobre líneas ADSL.

Existen dos modos de realizar estos test, *Out of the box*, donde solo es modificable el compilador empleado o *Full Fury* donde los fabricantes pueden modificar todo el algoritmo para aprovechar los recursos del elemento sometido a test y conseguir así una mayor optimización.

Todas las puntuaciones se calculan como la media geométrica de las evaluaciones individuales obtenidas en cada algoritmo. Luego se normalizan respecto a 307,455 que fue el menor valor que se obtuvo para un procesador embebido en diciembre de 2000 y el valor obtenido es la puntuación otorgada y publicada.

2.3.3. Linpack

El benchmark Linpack fue desarrollado en el *Argonne National Laboratory* por Jack Dongarra en 1.976, y es uno de los más usados en sistemas científicos y de ingeniería. Está formado por un conjunto de programas escritos en FORTRAN para la resolución de ecuaciones. Las operaciones que ejecuta son sumas y multiplicaciones de matrices en coma flotante con 64 bits –doble precisión- y sus resultados se expresan en MFLOPs (14).

La característica principal de Linpack es que hace un uso intensivo de las operaciones de punto flotante, por lo que sus resultados son muy dependientes de la capacidad de la FPU que tenga el sistema. Al realizar esencialmente cálculos con matrices es un test fácilmente paralelizable, y se puede utilizar para medir la eficiencia de sistemas multiprocesador, de hecho, el TOP500 de computadores está basado en una variante de este PdC.

Linpack consta de dos pruebas, Linpack 100 x 100 y Linpack 1000 x 1000:

- El Linpack 100 x 100 resuelve una matriz 100 x 100 de ecuaciones lineales simultáneas.
- El Linpack 1000 x 1000 resuelve una matriz 1000 x 1000 de ecuaciones lineales simultáneas. A diferencia de la prueba anterior, en esta sí está permitido utilizar algoritmos optimizados por el fabricante.

2.3.4. Dhrystone (MIPS)

Fue creado en 1984 por Reinhol P. Weicker de Siemens-Nixdorf Information System y pretendía medir el rendimiento de los sistemas basados en computadores. Dada la existencia de otro conjunto de pruebas destinadas a las aplicaciones de coma flotante, Whetstone, Weicker optó por centrar sus códigos en tareas con enteros. El nombre Dhrystone lo acuñó para crear un juego de palabras con el nombre del PdC de flotantes. La última versión publicada fue Dhrystone v.2.1 en 1988 y se mantiene hasta la fecha. Aunque algunos fabricantes aún emplean este conjunto de códigos, realmente Dhrystone ha sido desplazado por los ya citados EEMBC para procesadores empotrados.

Este PdC es de tipo sintético, esto es, se crea específicamente para medir una o más características de un sistema, procesador o compilador. Los PdC sintéticos pueden imitar a las aplicaciones del mundo real o bien ser totalmente artificiales, es por esta razón por lo que no son eficaces para evaluar el rendimiento completo de un sistema y se reservan solo para estudiar una característica concreta.

Los códigos que lo componen actualmente están escritos en lenguaje C y ejecutan un conjunto de operaciones matemáticas con enteros, trabajos con cadenas de caracteres, decisiones lógicas y accesos a memoria. Las ventajas de Dhrystone son su reducido tamaño, solo ocupa en memoria 1.5 KBytes, su carácter público y que es sencillo de ejecutar. Por ser tan pequeño el Dhrystone entra completamente en la caché interna, de esta manera no evalúa el resto del sistema pero presenta la ventaja de que mide solamente la capacidad del procesador para trabajar con enteros.

Al ejecutar los códigos que componen el Dhrystone, como valor de salida devuelve el número de iteraciones por segundo (Dhrystones por segundo) que ha realizado el procesador a una determinada frecuencia. El resultado debe compararse con una máquina de

referencia, la VAX 11/780, que corre a 1 MIP y logró 1.757 iteraciones por segundo. El valor normalizado queda expresados en millones de instrucciones Dhrystone por segundo (Dhrystone MIPS). Si los DMIPS calculados se dividen por la frecuencia del procesador bajo test, se obtiene una puntuación en DMIPS por MHz.

Como desventaja Dhrystone presenta la incapacidad para realizar operaciones en coma flotante, por lo que muchos fabricantes no lo consideran como una medida adecuada para definir el rendimiento de hoy en día. No existe un proceso de certificación para citar los resultados del Dhrystone. Esto puede ser una ventaja en cuanto a la facilidad de trabajar con los resultados, pero una desventaja al no quedar claro que versión se emplea o si se ha seguido el procedimiento correcto (15).

Los Dhrystones son utilizados para evaluar el rendimiento de los sistemas empleados para el desarrollo de aplicaciones. Los resultados de los PdC Dhrystone son publicados en la UNIX Review y otras publicaciones UNIX.

2.3.5. Whetstone

Predecesora del Dhrystone, es una medida desarrollada para evaluar sistemas destinados a la ejecución de pequeños programas científicos y de ingeniería. Sus códigos se desarrollaron en FORTRAN e incluyen cálculos con enteros y en coma flotante, manipulación de matrices y saltos condicionales.

Esta prueba predice cómo serán ejecutadas las aplicaciones que hacen un uso intensivo de la unidad central de proceso. Los resultados son expresados en KWIPS (miles de instrucciones

Whetstone por segundo). Los resultados de los Whetstone son publicados en la *UNIX Review* y otras publicaciones UNIX.

2.4. Iniciativas de PdC para FPGAs

2.4.1. PREP®

Durante los años 90 el número de fabricantes que ofrecían componentes de lógica programable, en adelante LP, como los dispositivos tipo FPGA, llegó a ser muy alto. Las herramientas existentes eran complejas e incompatibles entre ellas dificultando aún más la elección del LP más adecuado para un determinado diseño. La mayor parte de los diseñadores empleaban sus propios circuitos para hacer pruebas de rendimiento y capacidad y así determinar el modelo idóneo, era un trabajo duro, costoso y largo.

Para tratar de paliar los problemas citados, en octubre de 1991 tres empresas interesadas en el mundo de la lógica programable -dispositivos, diseños, herramientas y metodologías- formaron un consorcio denominado *Programmable Electronics Performance Corporation* (PREP®). Aunque PREP® inicialmente fue liderado por Xilinx Inc., Actel Corp. y Altera Corp., llegó a contar con hasta trece compañías del sector entre las que estaban Intel Corp., AMD Corp., AT&T, Quicklogic Inc., o Lattice Semiconductors Corp. entre otras.

La principal misión de este consorcio era promover el conocimiento de la LP y aportar un estándar que permitiera comparar entre los múltiples productos comercializados atendiendo al criterio de rendimiento y capacidad. El PdC propuesto trataba de ahorrar este trabajo a los diseñadores, simplificando la tarea de elección del dispositivo.

La organización se encargaba de las siguientes tareas:

- Desarrollo y mantenimiento de los circuitos que componían el PdC
- Determinación de una metodología de medida
- Publicación de resultados
- Verificación de las pruebas realizadas por los fabricantes
- Certificación de los productos estudiados con este PdC
- Proteger el estándar contra malos usos del mismo

Los circuitos elegidos para componer el PdC, un total de diez, estaban basados en aplicaciones ampliamente utilizadas. Y se trataba de que el diseñador identificara entre ellos algunas de las funciones que formarían su proyecto final. A continuación se detalla brevemente cada uno de estos circuitos:

CIRCUITOS
Camino de datos: compuesto por un multiplexor de 4 entradas y 1 salida, un registro y un registro de desplazamiento, todos ellos de 8 bits de ancho que gestionan la carga paralela de un registro de desplazamiento de ocho bits. Tres de los bytes son comunes a todas las instancias que de este PdC se haga y el cuarto procede de la instancia anterior.
Temporizador/contador: compuesto por dos registros, un multiplexor de 2 a 1, un contador y un comparador, todos de 8 bits.
Máquina de estado simple de ocho estados. Sus ocho salidas son función del estado actual y de los ocho bits de entrada.
Máquina de estado compleja de dieciséis estados y cuarenta transiciones con ocho bits de entrada y ocho de salida.
Circuito aritmético: multiplicador con signo de 4x4, su salida va hacia un acumulador que dispone de una señal de control para solo almacenar el resultado o sumarlo al contenido anterior.
Acumulador de 16-bits

Contador binario de 16-bits con precarga
Contador binario de 16-bits con pre-escalado. El contador no tiene que empezar la cuenta hasta varios ciclos de reloj después de que el valor haya sido cargado.
Decodificador de direcciones de 16-bits
Circuito de relleno

Tabla 3. Circuitos incluidos en el PdC de PREP

Para ejecutar el PdC, la organización PREP elaboró un método para la medida de capacidad y otra para la de rendimiento.

I. **Metodología para medir la capacidad lógica**: el dispositivo debe ser utilizado al máximo de su capacidad siempre que lo permitan las líneas de interconexión. Para conseguirlo se instanciará el circuito tantas veces como sea posible Figura 7. Las señales de reloj y de control llegarán por igual a todas las instancias, y las entradas externas solo al primer circuito cuyas salidas alimentarán al siguiente. Para mantener la independencia de los circuitos cada instancia conectará sus salidas a flip-flops. La implementación se realizará mediante herramientas automáticas guiándolas por objetivos, bien para conseguir el mínimo de área ocupada o bien para obtener la mayor frecuencia.

La máxima frecuencia que puede conseguirse en un dispositivo programable depende de la cantidad de recursos e interconexiones empleadas, por tanto, si la FPGA no está completamente utilizada, el resultado del PdC puede ser erróneo. Dado que algunos de los circuitos del PdC son suficientemente grandes como para solo permitir una instancia del mismo, el consorcio PREP aporta el circuito número diez denominado “de relleno”. Este circuito pretende ocupar los recursos que puedan quedar ociosos al incluir una única instancia del núcleo utilizado.

Este circuito se añade aislando sus entradas de las del circuito principal para que no participen en la frecuencia obtenida.

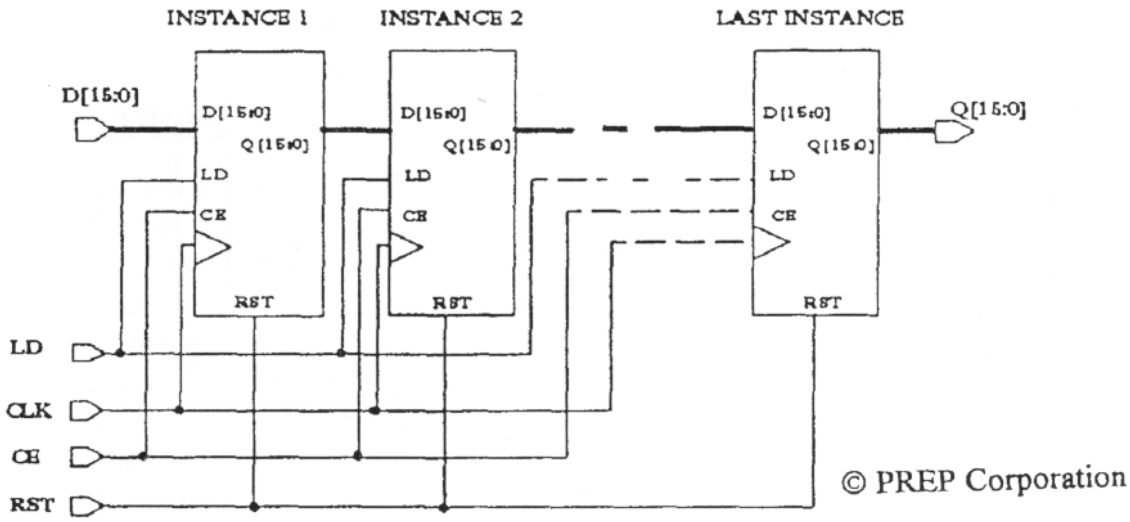


Figura 7. Esquema de cómo realizar n-instancias de cualquier circuito del PdC (Fuente:PREP©)

II. **Metodología para la medida de tiempos:** debe medirse tanto la frecuencia interna máxima a la que puede operar el dispositivo como la frecuencia a la que puede comunicarse con otros dispositivos externos.

La frecuencia interna máxima viene dada por la instancia del circuito más lento. Debe darse información sobre la mayor frecuencia conseguida, la menor y la frecuencia promedio de todas las instancias del circuito que se implementaron. La frecuencia máxima se mide como el inverso del mayor retardo de los siguientes caminos: el mayor camino entre un flip-flop de una instancia y el de la inmediatamente anterior, el mayor camino entre dos flip-flops de la misma instancia. Los retardos de los caminos siempre incluyen el clock-to-out del flip-flop de inicio, el retardo combinacional, el retardo de las

interconexiones y el tiempo de establecimiento (*setup*) del flip-flop de fin, Figura 8. Dado que estas medidas siempre dependen de la existencia de una instancia anterior, nunca deberán calcularse en la primera instancia.

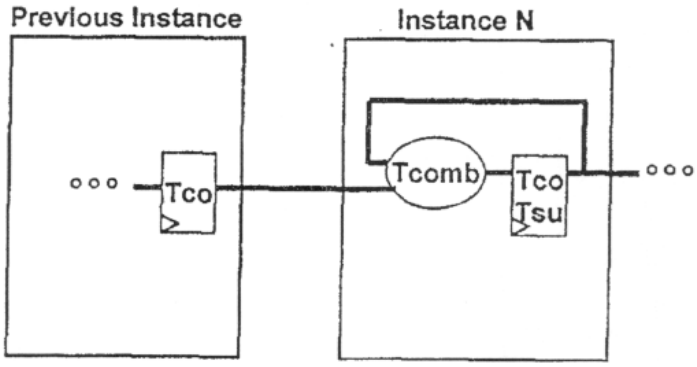


Figura 8. Modelo de una instancia y la inmediata anterior (Fuente:PREP©)

La frecuencia externa se define como la mayor frecuencia a la que puede comunicarse un dispositivo con otro exterior. Se calcula como la inversa del mayor retardo en los siguientes caminos: entre el flip-flop de la última instancia y el de la primera, o entre una señal de tipo global externa hasta el flip-flop de la primera instancia o de la última. Para el cálculo del primer camino debería conectarse la salida de la última instancia con la entrada de la primera, Figura 9.

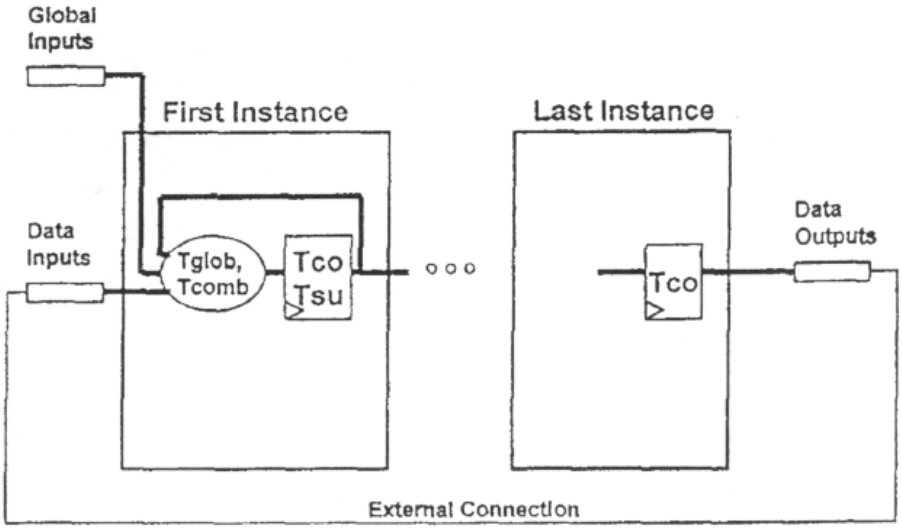


Figura 9. Conexión de la última instancia con la primera (Fuente:PREP©)

Una vez ejecutado el PdC se emitía un informe que incluía información sobre el porcentaje total de lógica ocupada en cada dispositivo, indicando además el número de instancias que se habían hecho del circuito y el total de elementos de relleno utilizados. Otro dato que debía figurar en el informe era el rendimiento obtenido, especificando las frecuencias máximas interna y externa conseguidas, con qué herramientas se había ejecutado el PdC y si se había realizado la implementación optimizando por área o por rendimiento.

PREP estableció las siguientes reglas para publicar los resultados:

1	Debe ejecutarse el PdC completo sobre cada dispositivo sobre el que quiera publicarse resultados.
2	Para todos los circuitos del PdC debe especificarse la máxima ocupación de área y los peores casos de frecuencia interna y externa.
3	Los datos publicados deben representar a los dispositivos con peores condiciones de temperatura, frecuencia y tensión.

4	Los datos deben ser obtenidos empleando herramientas de conexión y emplazamiento automáticas, en ciertos casos, puede hacerse alguna asignación manual, debiendo esta circunstancia detallarse en el informe.
5	Los fabricantes de LP pueden ofrecer resultados de todos sus productos, pero el de mayor tamaño siempre estará representado.
6	Solo se publicarán resultados de productos en venta
7	Los parámetros de tiempo y modelos de simulación deberán aparecer en las hojas de características y estar garantizados por los fabricantes.
8	Las herramientas empleadas en la ejecución de los PdC deben de estar disponibles para los usuarios.
9	Los resultados de los PdCs deben ser reproducibles por terceros. Las compañías integrantes de PREP verificarán los resultados antes de publicarlos.

Tabla 4. Reglas para la publicación de resultados de PREP

Los productos certificados por PREP recibían un sello y sus resultados eran publicados quedando a disposición de los usuarios con garantía de la organización. Existían dos sellos: el de “Certificado” y el de “Preliminar, No certificado”. Este último se acuñaba a los productos que habían sido sometidos al PdC por parte de los fabricantes y estaban a la espera de la verificación de PREP.

Mediante este PdC, PREP ofrecía una razonable estimación del total de celdas lógicas ocupadas y el rendimiento que podían esperar de distintos dispositivos programables. Sin embargo, eran conscientes de que se trataba de una visión limitada del dispositivo y no una medida perfecta de área y velocidad, realmente no trataban de determinar un ganador. El usuario debía tener en cuenta otra serie de factores que el PdC no ofrecía como la disponibilidad, consumo, reprogramabilidad, número de entradas/salidas o encapsulados entre otros (16) (17) (18).

En 1994 Altera propone un par de métricas, a partir del PdC PREP, que permitían establecer comparativas entre dispositivos

de forma rápida, ahorrando al usuario la lectura de los resultados de los nueve circuitos por cada modelo de PLD que quisiera estudiar. Define dos parámetros a los que denominó ABS (average benchmark speed) y ABC (average benchmark capacity) y que se calculan a partir de las ecuaciones siguientes:

$$\mathbf{ABS} = \frac{(F_{ext1} + F_{max-media1}) + \cdots + (F_{ext9} + F_{max-media9})}{18} \quad \textbf{(Ecuación 3)}$$

$F_{ext\ n}$ → Frecuencia máxima externa conseguida en cada circuito del PdC
$F_{max-media1}$ → Valor medio de las frecuencias internas de cada circuito

$$ABC = \frac{I1 + I2 + \cdots + I9}{9} \quad \textbf{(Ecuación 4)}$$

I_n → Número de instancias de cada uno de los nueve circuitos del PdC

2.4.2. Universidad de Massachusetts

En 2006 el grupo de investigación de "Computación reconfigurable" de la universidad de Massachusetts desarrolla y publica una colección de circuitos que permiten medir el rendimiento y capacidad de las FPGAs sobre las que se implementa. Los diseños están disponibles gratuitamente pero no pueden ser utilizados en aplicaciones comerciales.

Los diseños propuestos fueron adaptados de algunos circuitos pertenecientes al consorcio OpenCores y se diseñaron para estudiar y comparar el consumo de las FPGAs con microprocesadores. Se probaron con el dispositivo programable de Altera Stratix y el microprocesador ST230 VLIW de ST Microelectronics, ambas empresas colaboraron económicamente en este proyecto de investigación capitaneado por el profesor Russell Tessier.

El PdC está formado por los siguientes circuitos:

- Filtro FIR: filtro digital de respuesta finita, el número de orden es escalable y los coeficientes son configurables.
- Transformada directa discreta del coseno (FDCT)
- Codificador compresor JPEG
- ASOVA turbodecodificador/codificador: decodificador para la corrección de errores de alto rendimiento basado en el algoritmo Viterbi con salida soft (no devuelve valores 0 ó 1 sino un gradiente entre ellos) adaptativa (se adapta a los datos a comprimir).
- Decodificador de comunicaciones para la corrección de errores AVA, basado en el algoritmo Viterbi adaptativo
- Decodificador de comunicaciones para la corrección de errores adaptativo Reed-Solomon

Todos estos diseños han sido optimizados para usar todos los recursos empotrados de la FPGA de Altera y tratando de conseguir el máximo paralelismo. Además de proporcionar los ficheros Verilog o VHDL, también incluyen ficheros de test y el proyecto creado con la herramienta de Altera Quartus (19).

2.4.3. Tesis doctoral "Síntesis de circuitos aritméticos sobre FPGAs. Una evaluación cuantitativa" (20)

En esta tesis de 2003 realizada al amparo del Grupo de Investigación de Circuitos Microelectrónicos de la Universidad de Cádiz, el autor aborda un estudio sistemático de circuitos aritméticos, analizando la capacidad de las FPGA para implementar dichos circuitos al tiempo que se comparan diversas

arquitecturas y algoritmos, obteniendo criterios para seleccionar el más adecuado para cada tipo de operación estudiada y para cada familia de dispositivos empleada. Estos criterios son: recursos empleados, velocidad y tiempo de síntesis e implementación.

Los circuitos fueron descritos utilizando el lenguaje VHDL y se sintetizaron sobre las familias 4000, SpartanII y Virtex de Xilinx.

Los algoritmos incluidos en este banco de pruebas se recogen en la tabla Tabla 5.

Sumadores	Multiplicadores
Con propagación del acarreo	Suma y desplazamiento
Con anticipación del acarreo	De Wallace
Con selección del acarreo	De Booth modificado
Con salto del acarreo	
Con suma condicional	
Con acarreo condicional	
Multioperando con ahorro del acarreo	

Tabla 5. Circuitos del PdC de la tesis "Síntesis de circuitos aritméticos sobre FPGAs. Una evaluación cuantitativa"

Este PdC permitió analizar y evaluar los recursos especiales propios de las FPGAs pensados para soportar tratamiento aritmético: las líneas de propagación de acarreo.

2.4.4. Benchmarks de Altera Corporation

Ante la inexistencia de una propuesta estandarizada que permitiera establecer una comparativa entre las distintas FPGAs existentes en el mercado, en 2007 Altera Corporation establece

su propio PdC a partir de diseños reales de algunos de sus clientes. Opta por no crear circuitos artificiales o evaluar rendimientos con IPs aislados en el dispositivo. Los diseños que emplea son extraídos de distintos sectores como las telecomunicaciones, electrónica de consumo o comunicaciones inalámbricas.

Altera invierte una gran cantidad de recursos en estos PdCs y realiza sus pruebas con varias herramientas. Utiliza una aplicación de terceros denominada Synplify para las FPGAs de alto rendimiento y esta misma junto con la ofrecida por cada fabricante para comparar las FPGAs de bajo coste.

Destina un equipo de ingenieros a la adecuación de los diseños a la arquitectura bajo test para optimizarlos usando los recursos específicos de los dispositivos de Altera y Xilinx. No cuenta con FPGAs de otros fabricantes, y se limita a compararse con su principal competidor.

Altera presta especial interés a los parámetros de configuración de la herramienta que emplea y a las restricciones impuestas al diseño. Normalmente, las herramientas CAD ofrecen una serie de opciones por defecto que permiten implementaciones con un equilibrio en cuanto a tiempo, área, consumo y tiempo de compilación. Sin embargo, cada diseño puede optimizarse modificando estos parámetros convenientemente. En su método, Altera propone tres caminos:

- Mínima intervención del usuario y de la herramienta, para ello deben mantenerse las opciones por defecto. Los diseños no incluirán restricciones.

- Comparar estableciendo restricciones de tiempo agresivas asegurándose de que la herramienta está configurada para optimizar por rendimiento

- Tratar de conseguir los mejores resultados de tiempo y área. Implica máxima intervención del usuario y un amplio conocimiento del diseño a implementar así como de la herramienta. Tanto Xilinx como Altera ofrecen utilidades que permiten obtener diferentes resultados para un mismo diseño haciendo variaciones en algunos de los parámetros de implementación.

Para garantizar que los diseños son equivalentes para todas las FPGAs, Altera propone añadir código HDL al diseño original – le denomina envoltura- de modo que las funciones específicas que pudieran aparecer en el código original, fuesen interpretados por cualquier dispositivo. De este modo se evita la incompatibilidad originada en aquellos diseños que hacen uso de recursos hardware de un modelo concreto. El método que proponen para realizar la envoltura es:

- Determinar los módulos que emplean una estructura específica de una FPGA

- Rehacer el módulo para que pueda funcionar en la otra arquitectura a comparar

- Verificar que la funcionalidad del modulo rehecho es idéntica a la original

A la hora de interpretar los resultados Altera extrae información sobre:

I. Comparación de tiempos

Las métricas que se buscan son:

- Período mínimo o frecuencia máxima

- Tiempo de establecimiento para el pad de entrada

- Tiempo que transcurre desde que llega la señal de reloj al último flip-flop hasta que el valor aparece en el pad de salida

- Si hubo restricciones de tiempo, qué desviaciones aparecieron respecto de las medidas anteriores

Esta información se obtiene de los analizadores de tiempo que incluyen las herramientas distribuidas por los fabricantes de FPGAs y que se realiza de forma diferente en cada una. Establecer una comparativa justa es crucial debiendo verificarse que la información obtenida en cada herramienta es equivalente. Esto es, que ha tenido en cuenta las restricciones de los caminos, que mide el retardo de aquellos que no están restringidos, como trata las estructuras complejas de reloj y como calcula los valores de peor caso de establecimiento y mantenimiento de las entradas al dispositivo.

II. Comparación del uso de recursos lógicos

Si comparar las frecuencias de trabajo es una tarea difícil, comparar la lógica empleada no lo es menos. El principal problema radica en la diferencia existente entre las celdas lógicas básicas de cada arquitectura y en las características del resto de recursos que componen cada FPGA. Altera establece una relación entre las celdas lógicas básicas de cada uno de sus modelos con los equivalente de Xilinx, así por ejemplo, para el caso de las FPGAs de bajo coste establece que

1 Logic Element (LE) de Cyclone II de Altera = 0.5 Slice de Spartan3 de Xilinx **(Ecuación 5)**

Para diseños pequeños Altera devuelve información sobre el número total de elementos lógicos empleados (LUT, FF, MUX). Sin embargo, considera que para diseños grandes es mejor referirse a porcentaje de ocupación.

Los analizadores de tiempo devuelven información sobre el bloque de memoria empleado, sin embargo cada una lo expresa de forma diferente. Altera informa del total de bits que se ha empleado en cada implementación.

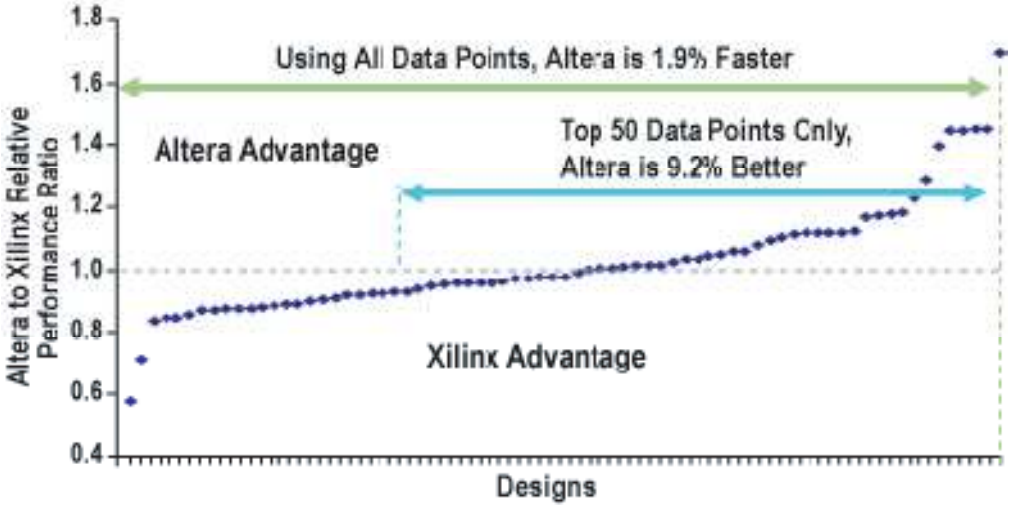


Figura 10. Comparación de rendimiento entre los dispositivos de Altera y Xilinx

De este PdC empleado por Altera no se conocen los circuitos que lo forman ya que están protegidos por pertenecer a terceras empresas. Los resultados obtenidos son ofrecidos solo por Altera sin que haya ningún otro organismo objetivo que los avale. En sus publicaciones aseguraron que en prom[1]edio la serie Stratix III era más rápida que las Virtex5 de Xilinx.

En cuanto a las FPGAs de bajo coste (21) concluyen resultados similares. El PdC es ejecutado con un menor número de circuitos

[1]

que para la FPGA de alto rendimiento, por cuestiones de capacidad, 49 en total. La mayor parte de ellos hacen un uso intensivo de los multiplicadores integrados en los dispositivos.

La fase de síntesis fue llevada a cabo nuevamente con Synplify y con la recomendada por cada fabricante de las FPGAs, Quartus II v4.0 SP1 e ISE 6.2i SP1. El emplazamiento e interconexiones en ambos estudios se realizaron con las aplicaciones de cada marca citadas.

Para la comparativa se emplearon dispositivos de Altera de bajo coste con el grado de velocidad mayor y el menor. De Xilinx se tomaron las FPGAs Spartan3 de mayor grado. Los resultados obtenidos por Altera se indican en la siguiente tabla:

Herramienta	**Dispositivos**	**Configuración Herramienta**	**Grado de velocidad**	**Relacion de rendimiento CycloneII/spartan3**
Synplify	Cyclone II Spartan 3	Best Effort Best Effort	Alto Alto	70,2%
Synplify	Cyclone II Spartan 3	Best Effort Best Effort	Lento Alto	29%
Quartus II ISE 6.2	Cyclone II Spartan 3	Best Effort Best Effort	Alto Alto	103,1%

Una de las razones aducidas por Altera para conseguir estas aplastantes ventajas sobre Xilinx, es que las entradas de sus LUTs presentan menores retardos que las de estos. Además cada una de las entradas tiene un retardo diferente pudiendo emplearse solo la entrada o entradas más rápidas en algunos diseños que no necesiten las cuatro.

Las conclusiones de Altera tras realizar estos PdC fue que era una tarea muy dura y compleja y que la metodología empleada influye fuertemente en los resultados finales. Algunos de los factores que más afectan son: los circuitos utilizados y su descripción, la configuración de las herramientas, las

restricciones impuestas al diseño y las diferencias entre los analizadores de tiempos (22) empleados en cada caso.

Conscientes de que la mejor forma de convencer a los clientes de la superioridad de sus productos era permitir que ellos mismos realizasen sus evaluaciones, plantean un nuevo PdC pero esta vez de uso público. Seleccionan los siete diseños más populares del consorcio OpenCores. Al tratarse de diseños abiertos Altera podía proporcionar a sus clientes todo el material necesario para ejecutar una comparativa fiable. Están disponibles para el público en general tanto los diseños RTL como sus restricciones y las configuraciones empleadas en las herramientas.

Los circuitos que forman el PdC son:

Nombre del circuito	**Función**	**Segmento**	**Lenguaje HDL**
Oc_usb_funct	Interfaz punto a punto de comunicación entre ordenador y periférico según estándar USB 2.0	Comunicaciones	Verilog
Oc_ethernet	Nivel MAC de ethernet que maneja el protocolo CSMA/CD de transmisión y recepcion de tramas cumpliendo la norma 802.1	Comunicaciones	Verilog
Oc_8051	Microcontrolador de 8 bits modelo 8051 incluye una serie de perifericos como temporizadores o contadores entre otros	Microprocesador	Verilog
OC_aquarius	CPU RISC que ejecuta instrucciones SuperH-2	Microprocesador	Verilog
OC_or1k	OpenRISC 1000: Arquitectura RISC abierta de 16/32 bits	Microprocesador	Verilog
OC_pci	Interfaz entre bus Wishbone y bus PCI local	Controlador de sistema	Verilog

Nombre del circuito	Función	Segmento	Lenguaje HDL
OC_des_des3perf	Algoritmo de encriptación Triple DES y DES	Encriptación	VHDL

La metodología propuesta para una correcta utilización del PdC (23) incluye:

- Instanciar el circuito tantas veces como permita el dispositivo y la herramienta.
- Añadir una envoltura que aúne todas las entradas y salidas de las instancias para reducir el total de pines utilizados, Figura 11 y Figura 12.
- Evitar caminos críticos entre la envoltura y el núcleo del circuito
- La envoltura añadida apenas influirá en el total de lógica empleada y tendrá su propio dominio de reloj.

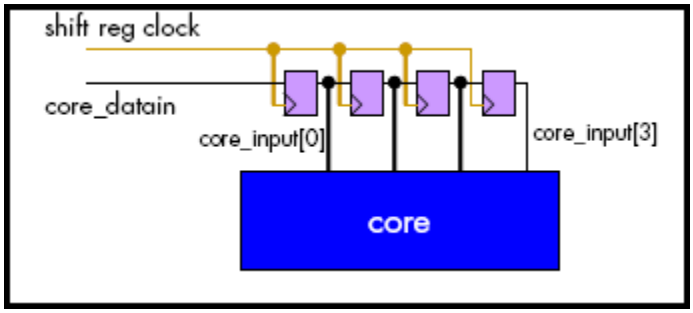


Figura 11. Registro de desplazamiento para las entradas de una de las instancias del circuito

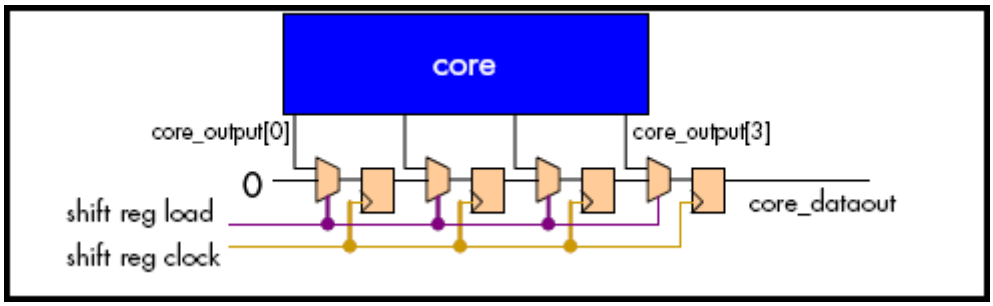


Figura 12. Registro de desplazamiento para las salidas de una de las instancias del circuito

Altera ha publicado sus resultados (24) y conclusiones tras ejecutar este PdC en las FPGAs de alto rendimiento, tanto en las propias como en las de Xilinx. Las herramientas empleadas fueron Quartus II v8.0 e ISE 9.2 SP4 para dispositivos de Altera y Xilinx respectivamente. Escogieron de cada fabricante los modelos con grado de velocidad media y similares características en cuanto a número de elementos lógicos.

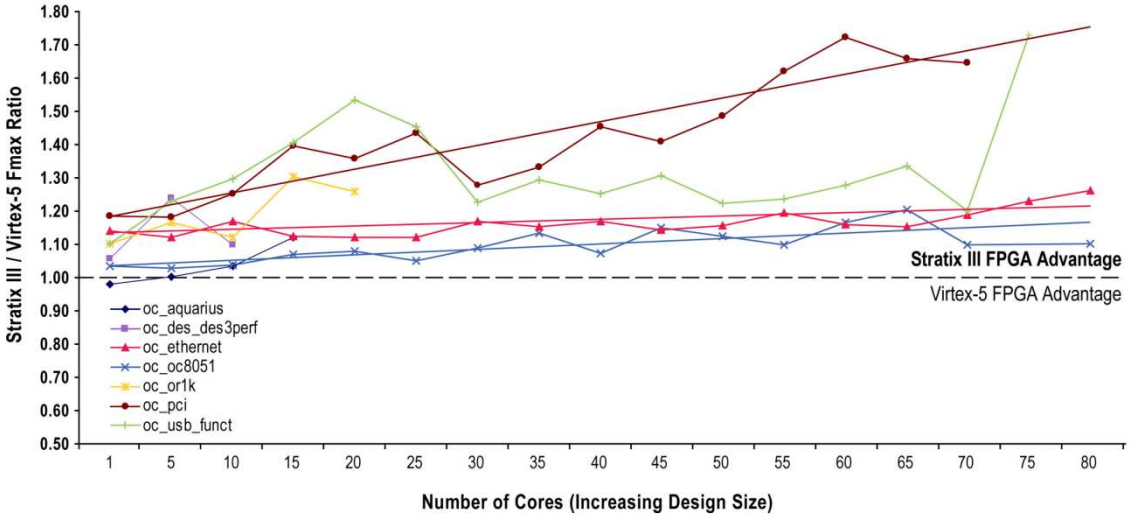


Figura 13. Relación de rendimiento entre Stratix-III y Virtex-5 para los siete circuitos de OpenCores

La Figura 13 muestra la relación entre la frecuencia máxima obtenida en una Stratix-III y una Virtex-5 (eje Y) en función del número de instancias empleadas (eje X). Hay una gráfica por cada uno de los siete circuitos que conforman el PdC. Todos los puntos situados por encima de la línea discontinua de valor 1, demuestra una ventaja de la FPGA de Altera sobre la de Xilinx y los que estén por debajo presentan el comportamiento inverso.

En cuanto a ocupación de recursos lógicos Altera vuelve a salir mejor parada, según sus evaluaciones, aventajando a las FPGAs de Xilinx en un 46% de media entre todos los diseños, Figura 14.

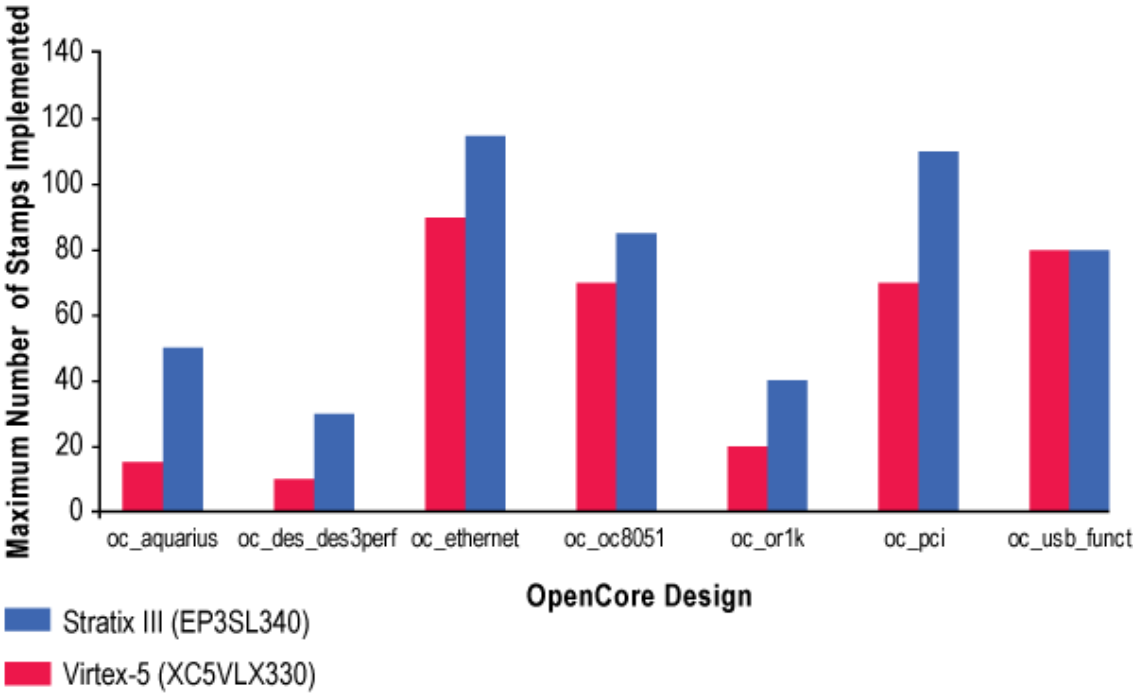


Figura 14. Máximo número de instancias posibles de cada circuito del PdC

Los tiempos de compilación de las herramientas vuelven a apuntar a la de Altera como la más rápida, QuartusII llega a superar a ISE en un factor de nueve.

Según Altera, la causa de que todas las pruebas indiquen a sus productos como los más eficaces radica en los módulos lógicos adaptativos (ALM) y la infraestructura de interconexiones denominada MultiTrack con que cuentan sus FPGAs de alto rendimiento. Presentan estos recursos como más óptimos frente a las LUT de de 6 entradas disponibles en la Virtex5. Los ALMs incluyen LUTs de 8 entradas, pero puede dividirse para incluir más de una función, algo imposible en las LUT de Virtex.

Aunque Xilinx no ha realizado ninguna propuesta de PdCs, sí que ha respondido a los resultados obtenidos por Altera. Defiende el rendimiento de sus productos ante este PdC señalando como injustas algunas consideraciones hechas por Altera:

- La herramienta de síntesis de Quartus II se emplea con opciones de mayor esfuerzo que la de ISE.
- La versión de ISE empleada es las 9.2i y aseguran que la disponible en ese momento era la versión 10.1i, incorporando ésta importantes mejoras frente a la anterior.
- Los diseños seleccionados por Altera no son totalmente objetivos, algunos están elegidas por hacer uso de abundantes registros, situación que beneficia a los dispositivos de Altera.
- El estilo de codificación y la implementación jerárquica de varias instancias del mismo núcleo dan lugar a una mala gestión de recursos tales como bloques de memoria o DSPs.

Xilinx repite las pruebas efectuadas por Altera pero esta vez haciendo sus propias consideraciones, especialmente en la configuración de sus herramientas y concluye que los resultados no les son tan desfavorables. Según ellos cuando sus dispositivos se rellenan al completo con varias instancias del

mismo IP, se consiguen un rendimiento 63% superior que con los productos de Altera, y además también son capaces de alojar un 3% más de instancias del mismo (25) .

2.5. Consideraciones

El repaso a los PdC de distintos tipo de sistemas, ofrece una clara idea de las dificultades que entraña la creación o elección de patrones suficientemente justos que permitan establecer una comparativa entre elementos similares. Por otro lado, tampoco es fácil determinar qué métricas pueden interesar más a cada usuario. Citando el grupo de Investigación de Ingeniería del Software de la *Lund University* de Suecia (12) la cuestión es: ***"¿Cuál es la pregunta a la que se responde con un determinado PdC?"***.

Cada diseñador que pretenda contar con dispositivos programables, ya sea con fines de investigación o de producción, tendrá una pregunta diferente que resolver. Atendiendo a los objetivos de este trabajo se determina que los sectores comerciales del campo de la electrónica sienten principalmente inquietud por dos factores críticos: el tiempo de mercado y el precio final del producto.

Vesey denomina tiempo de salida al mercado al "lapso de tiempo que transcurre entre la definición del producto y el momento en que se encuentra disponible en el mercado". Para productos tecnológicos, especialmente aquellos con un ciclo de vida corto, es un parámetro crítico que debe reducirse al mínimo si se pretenden grandes ventas. Disminuir el tiempo que conlleva la decisión de contar con un modelo u otro de lógica programable redundará en una mejora de este parámetro.

En el precio final del producto, el coste de los componentes del mismo es clave. Se debe elegir un elemento de lógica programable con capacidad suficiente para albergar el diseño, que cumpla los criterios de tamaño, forma, potencia y que tenga un precio unitario mínimo. A este precio habría que añadir el coste de la

herramienta software, del equipo programador y de la circuitería auxiliar según cada modelo.

Esta tesis propone un método para seleccionar e implementar módulos IP que permitan comparar lo más justamente posible un conjunto de FPGAs. Teniendo como referencia los trabajos anteriores, se añaden algunas novedades para responder mejor a las necesidades de los sectores de mercado que nos ocupan.

En contra de la idea de PREP o el primer planteamiento de Altera se ha preferido utilizar circuitos no sintéticos, de modo que los resultados sean lo más cercanos posible a la realidad.

Al igual que Altera, que modifica e incluye IP públicos, se opta por este mismo tipo de núcleos acudiendo al repositorio más popular: Opencores.

Frente al exhaustivo trabajo de todas las iniciativas que invierten gran cantidad de tiempo ajustando la configuración de las herramientas para conseguir un comportamiento exacto, se ha considerado interesante evaluar el binomio tecnología-herramienta en las condiciones de equilibrio ofrecidas por cada fabricante utilizando su versión gratuita. Siguiendo este método se cumple el objetivo de disponer de resultados referentes a la hora de llevar a cabo una implementación en el menor tiempo y sin necesidad de tener gran experiencia con la herramienta, además de deducir al precio final el de la herramienta.

Por último, este PdC no tratará de estresar los recursos tecnológicos con el objetivo de analizarlos en su peor caso, sino que utilizará la estrategia de implementar una única vez cada diseño consiguiéndose así su mejor rendimiento. Esta consideración es a su vez necesaria dado que no se modifica ninguna de las opciones de trabajo de la herramienta, y esto incluye la opción de eliminar la lógica redundante.

3. FPGAs DE BAJO COSTE

3.1. Introducción

Los sistemas electrónicos actuales se entienden como grandes bloques que incluyen microprocesadores, memoria, temporizadores, periféricos, etc., así como el software necesario para un completo control del mismo. Se les denomina sistemas en un chip, ó SOC de la voz inglesa *System On a Chip,* y los avances en integración han permitido tanto sus variantes programables, PSoC (*Programmable System On a Chip)*, como la posibilidad de implementarlos sobre FPGAs dando lugar a los FPSoC (*Field Programmable System On a Chip)*.

Las alternativas de implementación tecnológica para sistemas digitales electrónicos se han clasificado tradicionalmente de acuerdo a la Figura 15.

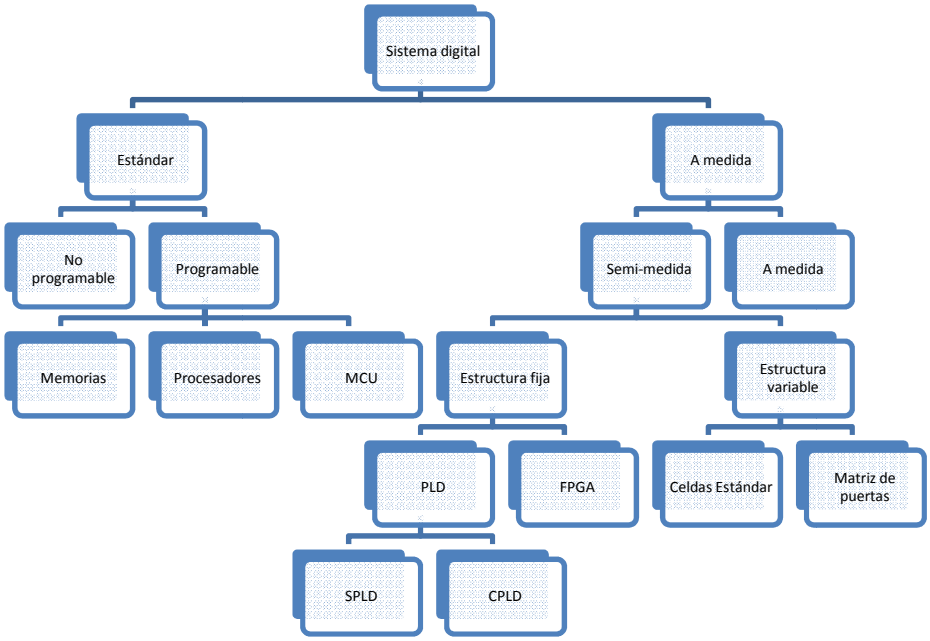


Figura 15. Clasificación de tecnologías de implementación de sistemas digitales

Sin embargo, esta clasificación trasciende de lo digital y en ocasiones algunos de sus ítems, ofrecen recursos analógicos como la FPGA Fusion de Actel.

Esta jerarquía sitúa a nuestras FPGAs como una alternativa al ASIC. Los circuitos a medida son siempre más baratos, de menor tamaño, mejor rendimiento y menor consumo que las FPGAs. Éstas al ser programables requieren de un mayor tamaño del dado del circuito, lo que da lugar a un mayor coste por unidad, presentan mayor consumo y son más lentas que los ASIC. Sin embargo, las FPGAs tienen como ventaja su reprogramabilidad por parte del diseñador, que da lugar a tiempos de llegada al mercado muy breves al lograr ciclos de diseño muy cortos.

Un ASIC requiere de un ciclo de varios meses para completar su fabricación y hay que tener en cuenta la posibilidad de errores que pueden ocasionar la repetición de la máscara y de los procesos de test, lo que supone un aumento de los costes y de tiempo. Una FPGA en 3 ó 4 semanas puede ser un producto final y no tendrá costes por error.

Un mismo modelo de FPGA puede usarse para distintos diseños, el ASIC una vez fabricado solo tiene una utilidad por tanto tiene un ciclo de vida menor. Además hay que considerar que el precio del ASIC empieza a ser realmente bajo cuando el pedido es de más de un millón de unidades y no todas la empresas necesitan solicitar estas cantidades, dependerá del tipo de producto final que piense comercializar.

El tamaño tanto de un ASIC como de una FPGA, a día de hoy, está condicionado por el número de terminales de entrada/salida que deba incluir y no por la cantidad de lógica del núcleo. A menor tamaño del dado más unidades del mismo caben por oblea dando lugar a una sensible reducción del precio. Así las FPGAs de bajo coste se han mantenido siempre en la vanguardia fabricándose en

procesos de 90nm ó 65nm con obleas de 12”, lo que ha dado lugar a que por ejemplo, las series Spartan de Xilinx hayan reducido su precio en un factor de 30 desde 1998.

Durante muchos años las FPGAs eran usadas principalmente para realizar prototipos de diseños que más tarde pasarían a ser ASICs y así obtener una gran ventaja en los costes por unidad al ser fabricación masiva. Hoy día su utilización es más amplia y la FPGA es el componente principal de muchos productos finales, especialmente en aquellos que no son de gran tirada. Según datos de la consultora Gartner, durante año 2009 los proyectos con FPGAs dominaron a los proyectos con ASICs en una relación de 30 a 1. Estos últimos decaen en un 22% siendo el segundo año consecutivo de descenso (13).

Casi cualquier producto de la electrónica de consumo, como un reproductor de MP3, tiene un ciclo de vida muy corto debido al pequeño plazo de tiempo en que el mercado le reclamará cambios y novedades. Por ejemplo menor tamaño, disponibilidad de radio FM o mayor capacidad de almacenamiento, lo que obligará al fabricante a retirar del mercado el dispositivo ya obsoleto y rápidamente presentar el nuevo modelo. Probablemente tampoco fabricó demasiadas unidades conocedor del ritmo trepidante de estas tecnologías. Como conclusión tuvo dos razones para no usar ASICs, una, la cantidad de unidades que debía pedir no amortizaba su coste, y dos, las pequeñas modificaciones para el nuevo modelo hubieran necesitado de un ASIC totalmente nuevo con todos sus inconvenientes.

En párrafos anteriores se ha explicado como las FPGAs han conseguido desbancar en algunos productos y mercados a los tradicionales ASIC, sin embargo, algunos autores creen que recientemente las FPGAs han recibido una dosis de su propia medicina, ya que las de bajo coste han sustituido a las de altas prestaciones en algunos casos. Este “canibalismo” ha dado lugar

a un estancamiento de los beneficios por ventas durante el año 2007.

Mucho tiempo ha pasado desde que en 1998 Xilinx pusiera en el mercado su primera FPGA de bajo coste Spartan. Este modelo se fabricaba usando tecnología mixta de 0,18 µm y 0,21 µm y seis capas de metal. Desde 2003 casi todas las FPGAs de bajo coste se fabrican en 90nm consiguiéndose precios muy competitivos adquiriendo lotes en torno a las 250.000 unidades. Puede decirse que 2003 fue el año en que las FPGAs de bajo coste consiguieron imponerse en el mercado. A día de hoy la mayor parte de los fabricantes continúan con los 90nm como densidad de fabricación en sus FPGAs de menor precio, aunque Altera ofrece las Cyclone III a 65nmn y anuncia Cyclone IV. Actel ha cerrado acuerdos con UMC para fabricar en 65nm y Xilinx ya vende la serie Spartan6 de 45nm fabricadas sobre obleas de 300mm.

3.2. Fabricantes

Los dispositivos programables de tipo FPGA comenzaron su andadura allá por el año 1985. Fueron varias las empresas que por entonces se fundaron para cubrir lo que entonces era un pequeño nicho del mercado de los semiconductores. Cuatro de ella ya están bien establecidas: Xilinx, Altera, Actel y Lattice Semiconductors.

A continuación se citan las principales empresas fabricantes de dispositivos lógicos programables, al citar sus productos se destacarán principalmente las FPGAs.

3.2.1. XILINX Inc.

Ross Freeman junto a Bernie Vonderscmitt y a Jim Barnett fundaron la empresa Xilinx en 1984. Actualmente tiene su sede central en San José, Estados Unidos, mientras que en Europa se hallan en Dublín (Irlanda) y en Asia en Singapur. En 1985 comenzaron a comercializar su primera FPGA: el modelo XC2064.

Xilinx desarrolla FPGAs y CPLDs que dan lugar a los tres grandes grupos que actualmente comercializa:

- Como electrónica adicional ó *glue logic* ofrece los CPLD de las familias *CoolRunner y CoolRunner II*, además mantiene algunos de los modelos más antiguos como la XC9500 entre otras.

- Para productos de alto volumen la económica familia Spartan. Se mantienen en el mercado algunos modelos de Spartan II, aunque actualmente los nuevos diseños optan por las potentes Spartan 3 y desde finales de 2009 es posible disponer de las renovadas Spartan6.

- Destinada al mercado más exigente la serie Virtex, caracterizada por su alto rendimiento y prestaciones. Las familias que la forman son Virtex-II, Virtex-5 y las recién lanzadas Virtex6 fabricadas con tecnología de 40nm.

- Por último cabe destacar algunos de los productos específicos para sectores especiales:

    - Automoción: Serie Spartan XA de precios muy reducidos que cumple las especificaciones de las normativa AEC-Q100.

- Aeroespacial y militar: específicas para entornos duros de trabajo o exposición a las radiaciones.

Xilinx Inc. ofrece a sus usuarios las herramientas necesarias para el diseño con sus componentes, documentación, soporte técnico, formación y una amplia gama de módulos software o IPs. Entre estos últimos se encuentra los microprocesadores de 8 y 32 bits, picoblaze y microblaze, que convierten a sus FPGAs, Spartan o Virtex, en un sistema en un chip con capacidad programable.

Entre algunos de sus principales clientes están Panasonic, Microsoft y Toshiba.

Para la fabricación de sus dispositivos, Xilinx utiliza una estrategia multiempresarial y cuenta con nombres diferentes según producto. En 2003 firmó un contrato con IBM (*International Bussines Machine*) en Estados Unidos y con UMC (*United Microelectronics Corporation*) en Taiwán para fabricar Spartan 3 con tecnología de 90nm sobre obleas de 300mm. En 2005 estableció la alianza con Toshiba en Japón para fabricar la Virtex 5 con tecnología de 65nm. La política multiempresarial garantiza a Xilinx el suministro aun en el caso de que alguna de las fábricas tenga problemas y esta seguridad se transmite hasta el comprador final. Por otro lado se aprovecha de la experiencia de fabricación de cada empresa según la tecnología a la que quiera migrar.

Año	Evento
2001	Fabricación en 150nm de la serie Spartan II
2002	Cambio a 130nm
2003	Con la fabricación a 90 nm sobre obleas de 300mm se reducen ostensiblemente los precios de la serie Spartan y su tamaño en un 80%. La nueva denominación es Spartan-3
2004	El precio de las Spartan se sitúa por debajo del de los ASIC estructurados. Xilinx lanza sus FPGAs para automoción, la serie Spartan3 XA

Año	Evento
	Xilinx vende un total de 1,5 millones de FPGAs
2006	Lanzamiento de la serie Spartan 3A optimizadas en I/O Protección del contenido de las FPGAs mediante el número de serie Device DNA Las ventas se sitúan en los 26 millones de unidades de dispositivos de 90nm, la serie Spartan 3 suponen el 24% de las ventas
2007	Aparece en el mercado las Spartan de tecnología FLASH, Spartan 3AN y las FPGAs especificas para tareas de procesado de señal, Spartan 3DSP
2008	FPGAs con DSP para automoción, Spartan 3A DSP
2009	Xilinx anuncia el lanzamiento de las renovadas Spartan6 con tecnología de 45nm

Tabla 6.Hitos destacables en la historia de las FPGAs de bajo coste de Xilinx

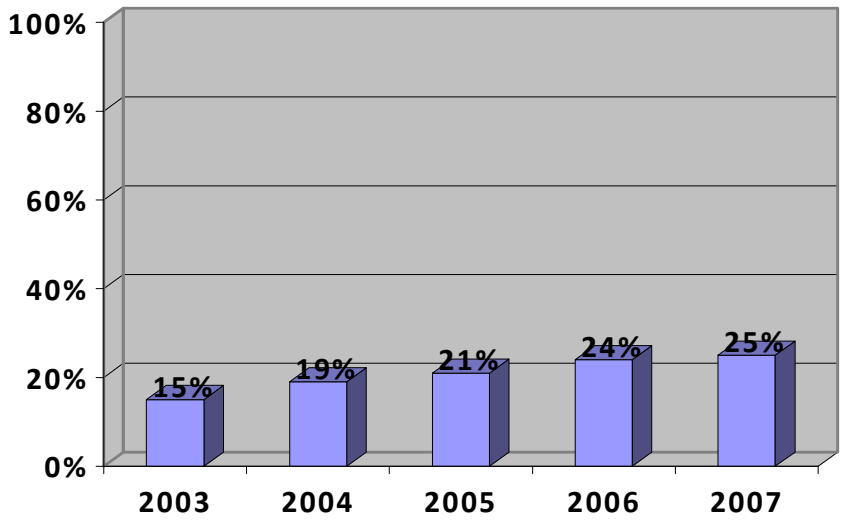


Figura 16. Porcentaje de ventas de Spartan3 sobre el total de LP de Xilinx

Xilinx Inc. cotiza en la bolsa americana de valores tecnológicos Nasdaq desde 1990, estando presente en el índice de las 100 empresas más representativas del sector tecnológico-industrial

denominado Nasdaq 100. En la tabla siguiente se ofrecen datos financieros y bursátiles de esta empresa:

	2008	2009
Valor de Mercado (nº acciones x valor/acción)	4.561 [2]	6.215
Beneficio neto por acción (EPS)	1,36	0,89
Price Earnings Ratio (PER)[3] (Precio acción/Beneficio acción)	12,13	25,3
Ingresos por ventas (millones) (FY 2008/2009)	1.841,372 $	1.825,18 $
Valor de cotización a 10-11-2008/2009	16,50$	22,52$
Valor Máximo de Cotización Fecha	91,87 $ Sep 2000	

Tabla 7. Valores bursátiles de Xilinx (XLNX)

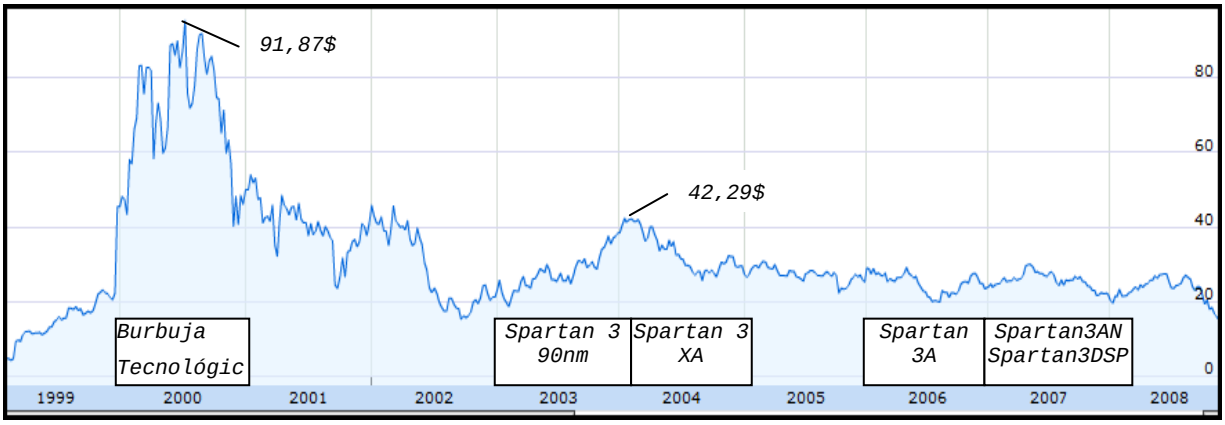


Figura 17. Evolución del valor de cotización de Xilinx desde el año 1999

3.2.2. ALTERA Corporation

Fue fundada en 1983 por Robert Hartmann, Michael Magranet, Paul Newhagen, y Jim Sansbury. En 1984 aparece en el mercado su

[2] En millones de dólares americanos.

[3] Índice que refleja la relación entre lo que cuesta la acción y el beneficio que presenta. El valor obtenido es el adecuado en el sector tecnológico que nos ocupa, según expertos.

primer dispositivo programable denominado EP300. Al igual que Xilinx, tiene su sede en San José, Estados Unidos y ofrece los mismos productos CPLDs, FPGAs, ASIC estructurados e IPs y también iguales servicios: software, formación y soporte técnico. En su cartera de clientes cuenta con conocidas empresas como Blaupunkt, Navman o Alcatel entre otras.

Altera ofrece los siguientes dispositivos programables de aplicaciones generales:

- Serie Cyclone con los modelos Cyclone, Cyclone II, Cyclone III y Cyclone IV. De bajo coste y orientadas a aplicaciones poco exigentes de productos de gran tirada.
- Serie Stratix con los modelos Stratix II, Stratix III, Stratix IV. Modelos de alto rendimiento que incluyen una gran cantidad de recursos hardware
- CPLD serie MAX, modelos MAX II y MAX. Económicos dispositivos de bajo coste y limitadas prestaciones.

Dispone de dispositivos de temperatura extendida para sectores específicos como el militar, automoción o el industrial.

Sectores	Rango de temperatura
Comercial	0°C to 100°C
Industrial	-40°C to 100°C
Extendido	-40°C to 125°C
Militar	-55°C to 125°C
Automoción	-40°C to 125°C

Tabla 8. FPGAs de temperatura extendida ofrecidas por Altera

Desde sus orígenes cuenta con TSMC (*Taiwán Semiconductor Manufacturing Company*) para fabricar sus productos optando por una estrategia de un único socio para esta labor. Con ellos ha

ido evolucionando desde el CPLD fabricado en 1988 con una densidad de integración de 0,65 µm, pasando por la FPGA Mercury de 0,18 µm en 2001, las primeras Cyclone con tecnología de 0,13 µm en 2002, con 90 nm en 2004 la Stratix II, hasta llegar hoy día a ofrecer una FPGA de bajo coste fabricada en 65 nm (Cyclone III) y en 45nm una de alto rendimiento (Stratix IV), las dos últimas se obtiene a partir de obleas de 300mm.

Año	Evento
2001	Introducción de la herramienta de desarrollo Quartus II
2002	Aparece la serie de bajo coste Cyclone con tecnología de 0,13 µm
2005	Lanzamiento de la serie Cyclone II con tecnología de 90 nm consiguiéndose una reducción de costes del 30%.
2007	Primera serie de bajo coste, bajo consumo y alto rendimiento con tecnología de 65nm, es la serie Cyclone III
2008	Nueva versión de la serie CycloneIII destinada al sector de la automoción
2009	Anuncio del lanzamiento de la serie Cyclone IV para el primer cuarto del año 2010

Tabla 9. Hitos destacables en la historia de las FPGAs de bajo coste de Altera

Al igual que Xilinx, Altera forma parte del grupo de empresas de referencia para el índice Nasdaq 100. En la tabla siguiente se resumen los datos más relevantes de carácter bursátil y financiero de esta empresa:

	2008	2009
Valor de Mercado (nº acciones x valor/acción)	4.590 [4]	6.127
Beneficio neto por acción (EPS)	0,99	0,78
Price Earnings Ratio (PER)[5] (Precio acción/Beneficio acción)	15,41	26,7

[4] En millones de dólares americanos.

[5] Índice que refleja la relación entre lo que cuesta la acción y el beneficio que presenta. El valor obtenido es el adecuado en el sector tecnológico que nos ocupa, según expertos.

	2008	2009
Ingresos por ventas (millones) (FY 2008/2009)	1.367,224 $	1.144,96[6] $
Valor de cotización a 10-11-2008/2009	15,26	20,83
Valor Máximo de Cotización Fecha	64 $ Agosto 2000	

Tabla 10. Valores bursátiles de Altera (ALTR)

Figura 18. Evolución del valor de cotización de Altera desde 1999

3.2.3. ACTEL Corporation

Fue fundada en 1985 y su sede central se encuentra en el estratégico enclave de la bahía de San Francisco, en concreto en Mountain View. Desde sus inicios, Actel ha ofrecido dispositivos programables con tecnología de programación basada en memorias Flash y conexiones antifusible. Esta característica ha dado lugar a que camine en solitario ya que la mayoría de productos del resto fabricantes emplean tecnología SRAM. La principal ventaja que ofrecen los productos de Actel es que además de reducir el coste por unidad, también consiguen bajar el coste

[6] Dato provisional, a la espera de que se publique el informe anual de 2009

del sistema, ya que al integrar memoria Flash no requieren de lógica adicional para almacenar la configuración.

Las FPGAs que comercializa actualmente son:

- Serie IGLOO de bajo consumo con los modelos IGLOO/e, IGLOO nano e IGLOO plus.
- La serie ProASIC Plus de bajo consumo y muy bajo precio por unidad incluye los modelos ProASIC/e, ProASICnano, ProASIC 3L.
- Además dispone de diversos modelos orientados a necesidades más especificas como:
    - Fusion, que incluye recursos analógicos programables
    - RTA-X tolerante a radiaciones
    - Las series antifusibles de alto rendimiento

A estos productos tangibles se le suman todos los servicios habituales en estas compañías: herramientas de desarrollo, amplio catálogo de IPs, soporte técnico, documentación y formación.

Actel ha establecido alianzas con diferentes empresas del sector de fabricación de circuitos tales como BAE Systems en USA, Matsushita en Japón y UMC en Taiwán. Con esta última acaba de firmar un acuerdo para fabricar dispositivos con tecnología de 65nm.

Los productos actuales de Actel pertenecen al nodo tecnológico de los 130nm, según las últimas noticias darán el salto a los 65 nm sin pasar previamente por los 90nm.

Año	Evento
2002	Lanzamiento de la serie de bajo consumo y bajo coste ProAsicPlus
2003	Series SX y MX destinadas a la automoción a precios desde 3$ por 250.000 unidades
2006	Aparece en el mercado la serie Igloo de bajo consumo (5µw)
2008	Lanza modelos de la serie ProASIC3 e Igloo por 99 centavos

Tabla 11. Hitos destacables en la historia de las FPGAs de bajo coste de Actel

Actel cotiza en el índice de valores tecnológicos americanos Nasdaq, aunque no forma parte del grupo Nasdaq 100 al no disponer de capital suficiente para ello. Los datos bursátiles más importantes de esta empresa son:

	2008	2009
Valor de Mercado (nº acciones x valor/acción)	243 [7]	320
Beneficio neto por acción (EPS)	0,11	0,25
Price Earnings Ratio (PER)[8] (Precio acción/Beneficio acción)	85,91	49
Ingresos por ventas (millones) (FY 2008/2009)	197,04 $	218,41 $
Valor de cotización a 10-11-2008/2009	9,45 $	12,25 $
Valor Máximo de Cotización Fecha	46,56 $ Jul 2000	

Tabla 12. Valores bursátiles de Actel (ACTL) a 10 de noviembre de 2008

[7] En millones de dólares americanos.

[8] Índice que refleja la relación entre lo que cuesta la acción y el beneficio que presenta. El valor obtenido es muy alto, no es comparable a Xilinx o Altera.

Figura 19. Evolución del valor de cotización de Altera desde 1999

3.2.4. Lattice Semiconductor

Fundada en 1985, aunque con sede central en Hillsboro (USA) cuenta con un centro de desarrollo en la capital de Silicon Valley, San José, junto al resto de empresas del sector. Su producto principal fue tradicionalmente el CPLD hasta 2002 cuando compró la división de FPGAs de Agere Systems Inc. Ese mismo año las ventas de FPGAs les supusieron el 12% de sus beneficios.

Lattice es pionera en la técnica ISP (*In-system-Programmability*) y XP (*Extended Programability*) para FPGAs, ambas marcas registradas, que permiten la reprogramación del dispositivo incluso ya colocado en la placa final.

Entre sus socios para fabricar las obleas se encuentran Fujitsu Microelectronics Limited y Seiko Epson ambas en Japón, United Microelectronics Corporation (UMC) en Taiwan y Chartered Semiconductor Manufacturing Ltd. en Singapore.

Ordena sus FPGAs en los siguientes grupos y como se observa aún no han superado el nodo tecnológico de los 90 nm:

- No-Volátiles y bajo coste
    - 90nm LatticeXP2
    - 130nm LatticeXP
    - 130nm MachXO PLDs
- Bajo coste
    - 90nm LatticeECP2
    - 90nm LatticeECP2M
    - 130nm LatticeECP
- Alto rendimiento
    - 90nm LatticeSC
    - 90nm Lattice FreedomChip Cost Reduction Path

A esta lista suma alguna LP de tipo GAL como MachXO orientadas fundamentalmente al bajo consumo y más en concreto a los dispositivos portátiles.

Año	Evento
2002	Adquiere la división de FPGAs de Agere Systems
2003	Lattice ECP con tecnología de 130nm. Consiguen un 18% de su beneficio por la venta de FPGAs
2004	Lanza las series LatticeECP-DSP™ y LatticeEC™
2005	Lattice XP de 130 nm de bajo coste y no volátil
2006	Lattice ECP2 con tecnología de 90nm sobre obleas de 300mm
2007	Lattice ECP2M que incluye un bloque SERDES (Conversión Serie/Paralelo para comunicaciones de alta velocidad)
2008	Lattice XP2 en 90nm pensado en altos volúmenes de producción

Tabla 13. Hitos destacables en las FPGAs de bajo coste de Lattice Semiconductors

Los datos bursátiles de esta empresa que cotiza en el NASDAQ, aunque no forma parte de las 100 de mayor capital, son:

	2008	2009
Valor de Mercado (nº acciones x valor/acción)	189 [9]	265
Beneficio neto por acción (EPS)	0,01	-0,07
Price Earnings Ratio (PER)[10] (Precio acción/Beneficio acción)	164	N/D
Ingresos por ventas (millones) (FY 2008/2009)	197,04 $	222,26 $
Valor de cotización a 10-11-2008/2009	1,64 $	2,30 $
Valor Máximo de Cotización Fecha	38,84 $ Sep 2000	

Tabla 14. Valores bursátiles de Lattice (LSCC)

Figura 20. Evolución del valor de cotización de Lattice Semiconductors desde 1999

Revisados algunos datos de las cuatro empresas más destacadas del sector de los dispositivos programables, cabe establecer un orden entre ellas. La consultora Gartner dio en abril de 2008 y de 2009 la siguiente clasificación, atendiendo a los ingresos de cada una durante los años 2006, 2007, 2008 y 2009.

[9] En millones de dólares americanos.

[10] Índice que refleja la relación entre lo que cuesta la acción y el beneficio que presenta. El valor obtenido es estratosférico.

Pos. 2006	Pos. 2007	Pos. 2008	Compañía	Ingresos 2006 (Mill. $)	Ingresos 2007 (Mill. $)	Ingresos 2008 (Mill. $)	% Cambio en ingresos 07-08	Cuota de Mercado 2008	Cuota de Mercado 2009
1	1	1	Xilinx	1.872	1.809	1.906	-3,4%	51,3%	**51,2%**
2	2	2	Altera	1.243	1.216	1.323	-2,2%	34,5%	**35,5%**
3	3	3	Lattice Semicon-ductors	245	229	222	-6,5%	6,5%	**6,0 %**
4	4	4	Actel	191	196	218	2,6%	5,6%	**5,9%**
6	6	5	QuickLogic	37	32	23	-13,5%	0,9%	**0,6%**
5	5	6	Cypres Semicon-ductor	34	28	21	-17,6%	0,8%	**0,6%**
7	7	7	Atmel	18	14	9	-22,2%	0,4%	**0,2%**
8	8	8	Chengdu SinoMicroe lec-tronics System	3	4	3	33,3%	0,1%	**0,1%**
			TOTAL	**3.643**	**3.528**		**-3,2%**		**100%**

Tabla 15. Ingresos por ventas de FPGAs/PLD y posición de cada empresa (Fuente: Gartner 2008/2009)

De los datos de la Tabla 15 se extrae como principal conclusión el dominio de Xilinx y Altera. Ambas empresas se beneficiaron del boom tecnológico de los 90, lo que les sirvió para establecerse en sus posiciones. En 1999 entre ambas sumaban el 63% de cuota de mercado de lógica programable y en 2009 han llegado al 85,8% (Fuente iSupli). Se evidencia que el resto de empresas no son competencia para estos dos gigantes aunque lo han intentado tomando caminos diferentes. Las firmas que ocupan los puestos del quinto al octavo, durante 2008 dejaron sus líneas de producción de FPGAs.

La empresa tercera y la cuarta, Lattice y Actel, han tratado de diferenciarse de las dos líderes y así conseguir su propia clientela. Se adelantaron ofreciendo FPGAs de configuración no volátiles, inicialmente usando técnicas antifusible y hoy día usando memorias Flash. Durante 2008 Actel empezó a ofrecer series de muy bajo consumo a precios tremendamente bajos, por debajo de un dólar para grandes pedidos. Sin embargo, ninguna de estas características ha dado los resultados esperados a efectos mercantiles. Se suman, como desventajas, que sus niveles de integración están estancados en 130nm, en el caso de Actel, y que sus frecuencias de trabajo son inferiores a las de Xilinx o Altera. De hecho, algunos de los grandes inversores de Actel creen que deberían replantearse sus estrategias y optar por prescindir de su línea de productos flash-130nm, ya que las pérdidas en 2008 fueron considerables.

Un análisis de los datos bursátiles nos revela algunas ideas interesantes:

A) Todas las empresas vivieron sus máximos de cotización durante el año 2000 cayendo a continuación estrepitosamente. La causa fue el boom de la burbuja tecnológica a veces denominada “burbuja.com”. En 1998 la bolsa americana vivió un ritmo trepidante, que se propagó al resto del mundo, debido principalmente a Internet. Muchas empresas fueron las que se

crearon para aprovechar la moda “virtual” y algunas de las tradicionales incluso segregaron actividades para resaltar cualquier cuestión relacionada con Internet (Caso de Telefónica con Terra en España). Entre estas empresas las había de muy diversas actividades, proveedores de servicio, comunicación, software, hardware, semiconductores, etc.

B) Tras la “explosión de la burbuja” y una vez estabilizado el mercado, todas alcanzaron sus máximos ya en condiciones normales, durante el año 2003. Sin embargo, ninguna vuelve a acercarse más a esos valores a pesar de que de 2003 a 2007 hubo fuertes subidas de la bolsa. Varias pudieron ser las razones que generaron dicha expectativa:

- Se pensó que las FPGAs sustituirían casi masivamente a los ASICs, el pronóstico hecho por distintas consultoras era que desde 2003 hasta 2008 sus ventas en general, crecerían en un 9%.

- Los nuevos procesos tecnológicos, la fabricación en obleas de 300mm con nivel de integración de 90nm, permitirían aumentar la producción y reducir los costes por unidad. Como conclusión, podrían ofrecerse mayores cantidades de FPGAs, con más funcionalidades y a precios mucho más bajos.

- A partir de 2004 casi todas las marcas introdujeron sus nuevas líneas de bajo coste con interesantes características a precios muy bajos y que bien podían haber reemplazado a muchos ASICs.

- Comenzó el despegue de la electrónica de consumo: reproductores de MP3, DVD, HDTV, telefonía móvil, etc.

- Las empresas lanzaron nuevas FPGAs destinadas a otro sector en auge como demandante de electrónica, la automoción.

- La introducción de características DSP en muchas FPGAs robaría mercado a los procesadores específicos de tratamiento de señal. La nueva televisión digital podría beneficiarse sustancialmente de estas mejoras.

- Asia, más en concreto China e India, demandarían gran cantidad de productos programables para emplearlos en el despliegue de las telecomunicaciones, internet y la electrónica de consumo.

Pero finalmente, aunque las ventas de LP crecieron, no lo hicieron del modo esperado, entre otras razones porque los ASICs fueron sustituidos principalmente por ASSPs (*Application Specific Standard Product*) más que por FPGAs, y Asia no fue el comprador esperado. Todo esto queda reflejado en los índices de cotización en bolsa de las empresas que nos ocupa. Es interesante comprobar la tendencia alcista que durante ese período tuvo otro índice, el de referencia de semiconductores en América (*Semiconductor Philadelphia Index*, SOXX).

Figura 21. Evolución del Índice de semiconductores de Filadelfia desde 2003. (Fuente: Finance Yahoo)

C) Xilinx y Altera son las empresas más sólidas y rentables de la lista, y destacan de las otras por:

- Ser mayores en tamaño y más fuertes económicamente
- Cotizan en Nasdaq100 dado su elevado valor de mercado
- Reparten dividendos entre sus accionistas
- El PER (*Price Earnings Ratio*) en ambas empresas se encuentra en el intervalo considerado normal, entre 12 y 15. Sus valores se aproximan mas al de empresas maduras (energía, transporte, servicios financieros, etc.) que a las consideradas en crecimiento (que invierten mucho en I+D a la espera de beneficios futuros).

D) Entre Xilinx y Altera hay poca diferencia a efectos bursátiles aunque Xilinx mantiene una posición ventajosa de ventas y beneficios respecto a Altera (51,3% contra el 34,5% en 2007). Sin embargo, esta última ha presentado en los últimos cinco años un porcentaje medio de crecimiento en ventas del 11,17% y del 16,93% de los ingresos netos. Esos valores son sensiblemente mayores que los de Xilinx establecidos en un 7,13% y un 5,41% respectivamente. A continuación se presenta una comparativa entre la evolución de los valores de cotización de las acciones de Altera, Xilinx, Actel y Lattice tomando como referencia el valor de cierre del 5 de diciembre de 2003. Se aprecia en el gráfico que todas las empresas han bajado sus valores desde 2003, siendo la línea de Lattice la que marca un descenso más pronunciado con un porcentaje del -73,92% aproximadamente,

Figura 22. Comparativa entre Xilinx, Altera, actel y Lattice desde 2003 (Enero 2010)

3.3. Aplicaciones de las FPGAs de bajo coste

Las FPGAs de bajo coste han evolucionado hasta incluir una gran variedad de recursos en su interior como multiplicadores, memorias, bloques DSP, interfaces de comunicación de alta velocidad, etc. Estas mejoras, manteniendo un interesante precio, así como un bajo consumo, han dado lugar a que distintos mercados cuenten con ellas en sus productos.

A los recursos embebidos en las FPGAs se le suma las ventajas que ofrece disponer de bloques software (IP) de todo tipo incluyendo microprocesadores de 8 o 32 bits. Todos los fabricantes de FPGAs ofrecen de forma gratuita núcleos de este tipo:

- Xilinx: ofrece MicroBlaze, núcleo de un microprocesador de 32 bits y picoBlaze de 8.
- Altera: su núcleo procesador es de 32 bits y se llama Nios II, su variante Nios II/e está especialmente diseñada para ser usado en sus FPGAs

de bajo coste, ya que hacen un menor uso de recursos lógicos del dispositivo.

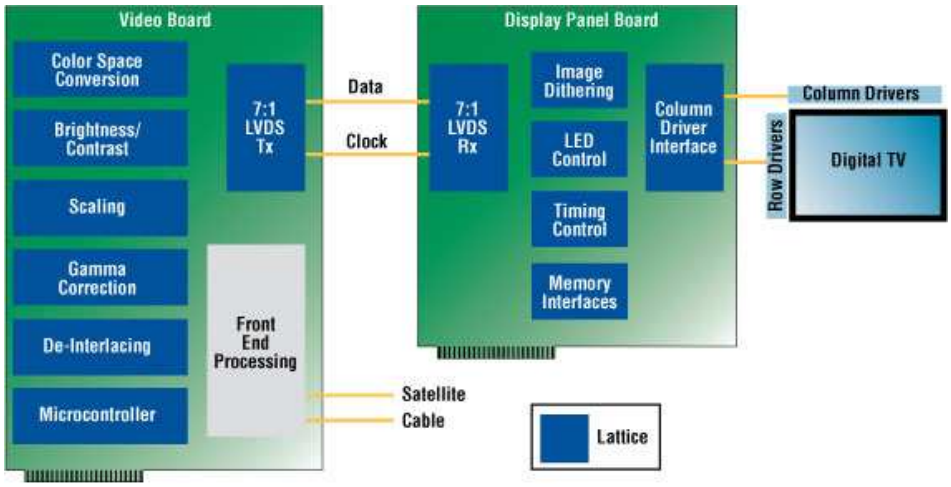
- Actel: tiene establecido acuerdos con ARM y sirve dos IP con los procesadores CortexM1 y CoreMP7 ambos de 32 bits y optimizados para implementar en sus FPGAs de bajo coste.
- Lattice Semiconductors: LatticeMico32 y LatticeMico8 son los soft-cores de 32 y 8 bits respectivamente que ofrece esta compañía, al igual que las anteriores tienen arquitecturas RISC.

Gracias a todas las características citadas se encuentra FPGAs de bajo coste en una infinidad de productos. A continuación se citan una pequeña parte de ellos:

- Imagen y sonido:
  - Cámaras de vigilancia: Sanyo cuenta con Spartan3 de Xilinx
  - Cámaras de video: Panasonic cuenta con Cyclone III de Altera debido a su óptima relación precio/rendimiento en procesamiento, y ofrece fácil solución mediante IPs para comunicarse con las tarjetas de memoria y control de visualizadores.
  - Visores de multimedia: Seiko Epson cuenta, con la FPGAs ProAsic3 de Actel dado su bajo consumo, reducido tamaño y por su interfaz de alta velocidad válido para comunicarse con tarjetas CompacFlash.
  - Televisores de plasma y LCD: Lattice ofrece IPs de funciones optimizadas para usar con LatticeECP2M en aplicaciones de televisión, tales como conversión

del espacio de color, ajuste de brillo y contraste, escalado, corrección gamma o conversión de entrelazado a progresivo, entre otros. Incluso ofrece el interfaz 7.1 LVDS para serializar/deserializar la comunicación entre la tarjeta de video con el monitor.

- Otros como: reproductores de DVD, grabadores personales (PVR), consolas de video juego, receptores de televisión satélite.



- **Figura 23. Esquema de bloques del circuito de control de un televisor digital ofrecido por Lattice**

- Automoción
  - Sistemas de navegación
  - Sistemas de ayuda al conductor: aviso de cambio involuntario de carril, control de crucero, monitorización de la presión de ruedas, etc.
  - Equipo de sonido, conexión a Internet, DVD, implementación de los distintos buses CAN, MOST, LIN ó FlexRay mediante IPs.

- Telecomunicaciones

    - Internet y redes: routers, conmutadores, MODEM-adsl, etc

    - Telefonía móvil: tanto terminales como las estaciones base. Estas últimas requieren de continuas actualizaciones y son grandes beneficiadas de la capacidad reprogramable incluso de forma remota de las FPGAs. Además no es un producto de altos volúmenes de producción por tanto no pueden justificar una inversión en ASIC o ASSPs.

    - Redes inalámbricas: dispositivos inalámbricos bluetooth o wifi.

3.4. Arquitectura de las FPGAs de bajo coste

3.4.1. Generalidades

De forma general todas las FPGAs cuentan en su arquitectura con un conjunto de bloques lógicos programables dispuestos en cuadrícula y alrededor de la misma se disponen los bloques de entrada y salida que conectan a la FPGA con el exterior. Si cuentan con recursos adicionales como bloques de memoria, o multiplicadores, estos se suelen encontrar entre las celdas lógicas. Todos los elementos se encuentran conectados por una compleja jerarquía de conexiones programables.

Los bloques lógicos suelen incluir una parte combinacional capaz de implementar funciones lógicas booleanas y de una parte

secuencial que permite sincronizar sus salidas con un reloj externo además de servir como registros, es habitual encontrar también lógica de acarreo. La parte combinacional varía de un fabricante a otro siendo las dos implementaciones más habituales las que a continuación se citan:

- Basada en tablas de búsqueda (*look up table*, LUT): se trata de una memoria SRAM que almacena una tabla de verdad. Con las entradas de la tabla se compone la dirección de cada celda que guarda el valor lógico de la salida de la función implementada correspondiente a esa combinación. Se puede implementar cualquier función lógica de n-entradas. Diversos estudios llegaron a la conclusión de que el valor óptimo de n era cuatro, con ese valor la penalización en área y conexiones no era demasiado elevada.
- Basada en multiplexores: requiere menos lógica y por tanto ocupa menos área que el modelo anterior pero en contrapartida no se pude implementar cualquier función lógica, deben adecuarse a las que establecen los multiplexores de la celda.

También es importante destacar las distintas tecnologías existentes para crear las conexiones entre los canales que unen todos los elementos de la FPGA, es lo que se denomina tecnología de fabricación. Las técnicas existentes en el mercado de las FPGAs de bajo coste son:

- SRAM: la configuración de las conexiones se almacena en RAM estática incluida en el chip de la FPGA. La memoria almacena la información necesaria para configurar adecuadamente los transistores de paso, puertas de transmisión o multiplexores que

forman los elementos de programación de las conexiones. Por tratarse de una memoria de naturaleza volátil su contenido se vacía al desconectar la alimentación del circuito, se hace por tanto necesario contar con un elemento externo, una ROM o una EEPROM, que guarde la información binaria de configuración de forma permanente. Al encender el circuito la ROM transferirá su contenido a la SRAM y entonces la FPGA comenzará a funcionar. Aunque son rápidas de programar tienen la desventaja de que es difícil de proteger el diseño que contienen.

- Flash: aportan sobre la anterior la ventaja de un menor tamaño y la no volatilidad, así las FPGAs se convierte en una solución monochip. Como desventaja presenta una mayor lentitud a la hora de ser programada. En cuanto a la protección del diseño son muy eficaces, ya que suelen incluir la posibilidad de almacenar una clave de hasta cientos de bits necesaria para poder leer o escribir datos del dispositivo.

A continuación se detallan las principales FPGAs de bajo coste existentes en el mercado.

3.4.2. Xilinx: Spartan3

Dentro de esta familia existen diferentes series orientadas a los mercados con alto nivel de producción y muy sensibles a los costes. Todas ellas se fabrican con tecnología CMOS con nivel de integración de 90nm, su tecnología de programación está basada en celdas SRAM y cuentan con algunas características comunes, a

efectos de bloques constructivos, que a continuación se citan. Por cada serie se detallará los elementos adicionales más importantes de cada una.

3.4.2.1 Arquitectura

Bloques lógicos configurables o CLBs (*Configurable Logic Block*) organizados matricialmente. Cada CLB está compuesto por cuatro partes denominadas slices que a su vez contienen dos celdas lógicas o LCs (Logic Cell). Cada celda lógica contiene una LUT de cuatro entradas, un biestable, dos multiplexores y lógica aritmética específica.

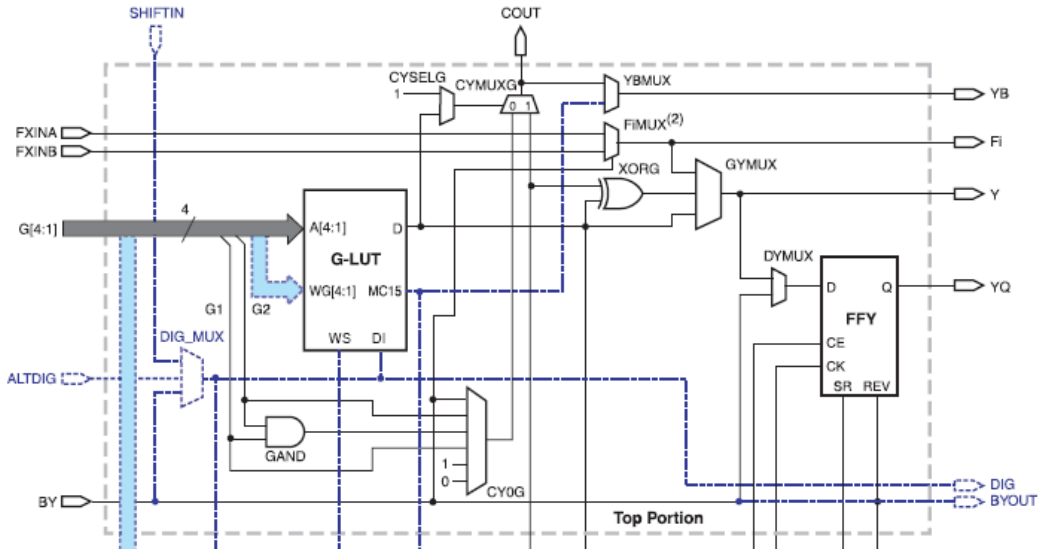


Figura 24. Detalle de la unidad lógica básica de la serie Spartan3 de Xilinx. (Fuente: Xilinx)

Cada CLB contiene dos tipos de slices, el SliceL y el SliceM. La diferencia principal entre ambas radica en la LUT, que en el SliceM puede también configurarse como memoria distribuida o como un registro de desplazamiento de 16 bits.

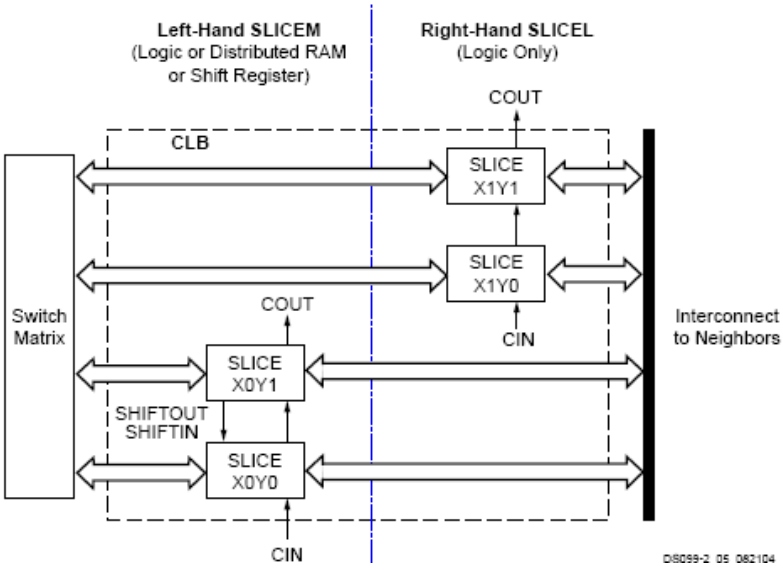


Figura 25. Estructura de un CLB de Spartan3 (Fuente: Xilinx)

Bloques de Entrada/Salida ó IOBs proporcionan la interfase entre los pines externos y la lógica interna del integrado. Cada IOB controla un pin que puede ser configurado como entrada, salida o bidireccional. El total de IOBs se divide en ocho grupos denominados bancos y que pueden configurarse de forma independiente variando ciertos parámetros como:

- Salida triestado
- Pendiente de cambio de la salida (Slew Rate)
- Nivel lógico de salida simple, hasta 18 tipos incluyendo LVTTL, LVCMOS, HSTL, SSTL
- Nivel lógico de salida diferencial, hasta 8 tipos incluyendo LVDS, RSDS, mini-LVDS, HSTL/RSSL
- Almacenamiento de la salida mediante un biestable
- Retardo programable
- Interfaz para DDR,DDR2 y SDRAM

- Resistencia de carga de salida (Pull-up/Pull-down)

Bloques de memoria de 18 kbits, el número de bloques depende del modelo. Pueden organizarse de distintas formas (ancho x profundidad) y configurarse como de puerto simple o doble. Se encuentran dispuestos en columnas ubicadas entre los CLBs.

Gestión digital de reloj o DCM encargados de eliminar el skew y de obtener desfases de reloj de alta resolución basados en elementos de retardo (DLL). El número total de DCM varía con cada modelo.

Multiplicadores que aceptan dos números binarios de 18 bits cada uno. El número de multiplicadores varía según modelo. En algunos, aquellos destinados al procesamiento de señal, los multiplicadores se sustituyen por bloques DSP48A que multiplican números de 18 bits e incluyen un acumulador de 48 bits.

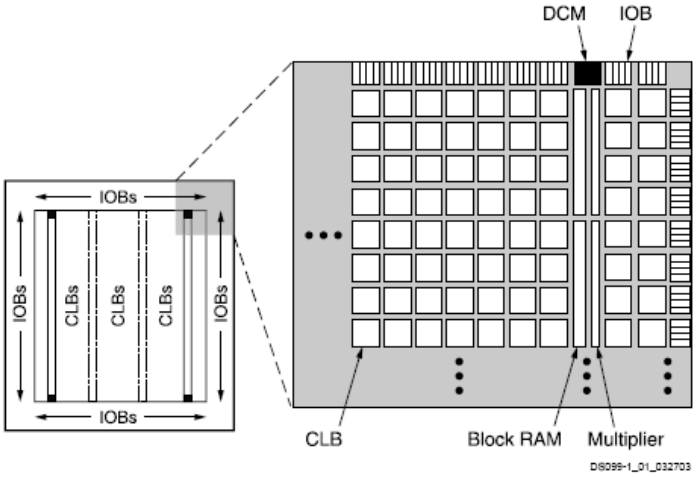


Figura 26. Arquitectura de la familia Spartan3 (Fuente: Xilinx)

Tradicionalmente Xilinx denominó celda lógica (LC) al conjunto formado por una LUT y un biestable. Usaba este elemento como referencia para indicar el tamaño de sus dispositivos. Actualmente puede decirse que su unidad lógica mínima es el *slice* que cuenta con más elementos que el antiguo LC, por eso mediante pruebas de evaluación ha establecido la siguiente equivalencia:

1 Slice = 2,25 Logic Cell (LC)

Por otro lado el nombre de cada dispositivo incluye las miles de puertas a las que equivale esa FPGA si se fabricara con un ASIC. Sin embargo no hay una relación clara entre ese valor y los LC.

3.4.2.2 Series Spartan3

Han evolucionado a través de las siguientes subseries

- **Spartan3**: Diseñada a partir de la serie SpartanII-E con mejoras funcionales derivadas de la tecnología Virtex-II.

Modelo	Puertas	Celdas Lógicas	Total de CLBs			Bits de RAM distribuid	Bloque de RAM	Mult.	DCMs	I/O	I/O dif.
			Filas	Columnas	Total						
XC3S50	50K	1.728	16	12	192	12K	72K	4	2	124	56
XC3S200	200K	4.320	24	20	480	30K	216K	12	4	173	76
XC3S400	400K	8.064	32	28	896	56K	288K	16	4	264	116
XC3S1000	1M	17.280	48	40	1.920	120K	432K	24	4	391	175
XC3S1500	1.5M	29.952	64	52	3.328	208K	576K	32	4	487	221
XC3S2000	2M	46.080	80	64	5.120	320K	720K	40	4	565	270
XC3S4000	4M	62.208	96	72	6.912	432K	1.728K	96	4	633	300
XC3S5000	5M	74.880	104	80	8.320	520K	1.872K	104	4	633	300

Tabla 16. Resumen de características de las Spartan3

- **Spartan3E**: diseñada a partir de la anterior pero incrementa la cantidad de lógica por cada I/O reduciéndose el coste por celda. El total de CLBs no se corresponde con el producto de filas y columnas debido a que en la matriz se incluyen también DCM, bloques de memoria y multiplicadores. En cuanto a los IOB, la serie 3E añade algunos que son solo de entrada e incorpora un retardo programable en todos además de permitir que los flips-flops DDR sean compartidos por bloques adyacentes.

Modelo	Puertas	Celdas Lógicas	Total de CLBs				RAM distribuida	Bloque de RAM	Mult. Dedic.	DCMs	I/O	I/O dif.
			Filas	Columnas	Total CLB	Total Slices						
XC3S100E	100K	2.160	22	16	240	960	15K	72K	4	2	108	40
XC3S250E	250K	5.508	34	26	612	2.448	38K	216K	12	4	172	68
XC3S500E	500K	10.476	46	34	1.164	4.656	73K	360K	20	4	232	92
XC3S1200E	1200K	19.512	60	46	2.168	8.672	136K	504K	28	8	304	124
XC3S1600E	1600K	33.192	76	58	3.688	14.752	231K	648K	36	8	376	156

Tabla 17. Resumen de características y precio de las Spartan3E

- **Spartan3A Extendida**: incrementa el número de I/O por lógica, reduciéndose el coste por I/O. Pensadas para altos volúmenes de producción y mínimo coste del sistema. Integran características adicionales como:

  - Empleo únicamente de dos niveles de tensión, elimina elementos de la placa bajando su precio final.
  - Seguridad con *DeviceDNA* que minimiza el riesgo de copia

  Dentro de este grupo existen tres series: Spartan3A, Spartan3AN, Spartan3AN-DSP.

Modelo	Puertas	Celdas Lógicas	CLB	Slice	RAM dist.	Bloq RAM	Flash bits[11]	Mult.	DSP48A	DCM	Maximum User I/O
XC3S50A/AN	50K	1.584	176	704	11K	54K	1M	3		2	144[12]
XC3S200A/AN	200K	4.032	448	1.792	28K	288K	4M	16		4	248[13]
XC3S400A/AN	400K	8.064	896	3.584	56K	360K	4M	20		4	311
XC3S700A/AN	700K	13.248	1.472	5.888	92K	360K	8M	20		8	372
XC3S1400A/AN	1400K	25.344	2.816	11.264	176K	576K	16M	32		8	502
XC3SD1800A	1800K	37.440	4.160	16.440	260K	1.512K	-	-	84	8	519
XC3SD3400A	3400K	53.712	5.968	23.872	373K	2.268K	-	-	126	8	469

Tabla 18. Resumen de características de las Spartan3A-Extendida

La serie Spartan-3AN incluye memoria de tipo Flash que puede almacenar la configuración o bien ser utilizada por el usuario, convirtiéndose en una solución con un único circuito integrado.

La serie Spartan-3ADSP reemplaza los multiplicadores dedicados por bloques DSP optimizados para conseguir velocidades de 250MHz en su menor grado de velocidad y un rendimiento de 30GMACs.

- **Spartan3-XA**: pensadas específicamente para el sector de la automoción: información, entretenimiento y asistencia al conductor. Disponibles en temperaturas de -40ºC a 125ºC, Q y de -40ºC a 100ºC, I) cumplen el estándar AEC-Q100.

[11] Solo disponible en modelos AN

[12] Máximo número de I/O de usuario 108 para la versión AN

[13] Máximo número de I/O de usuario 195 para la versión AN

Presentan tres líneas en esta serie: 3E, 3A, 3ADSP.

Modelo	Puertas	Celdas Lógicas Equiv.	F	C	Total CLBs	Slices	RAM Distribuida	Bloques RAM	Mult.	DCM	Maximum User I/O	Max. dif. Pares I/O
XA3S100E	100K	2.160	22	16	240	960	15K	72K	4	2	108	40
XA3S250E	250K	5.508	34	26	612	2.448	38K	216K	12	4	172	68
XA3S500E	500K	10.476	46	34	1.164	4.656	73K	360K	20	4	190	77
XA3S1200E	1200K	19.512	60	46	2.168	8.672	136K	504K	28	8	304	124
XA3S1600E	1600K	33.192	76	58	3.688	14.752	231K	648K	36	8	376	156

Tabla 19. Resumen de características de la serie SpartanXA-3E

Modelo	Puertas	Celdas Lógicas Equiv.	CLB				RAM distrib. bits	Bloq. de RAM bits	Multiplic. Dedic.	DCM	Maximum User I/O	Max. dif. Pares I/O
			F	C	Total CLBs	Total Slices						
XA3SD1800A	1800K	37.440	88	48	4.160	16.640	260K	1512K	84	8	519	227
XA3SD3400A	3400K	53.712	104	58	5.968	23.872	373K	2268K	126	8	469	213

Tabla 20. Resumen de características de la serie SpartanXA-3ADSP

Modelo	Puertas	Celdas Lógicas Equiv.	F	C	CLBs	Slices	RAM distrib. bits	Bloq. de RAM bits	Multiplic. Dedic.	DCM	Maximum User I/O	Max. dif. Pares I/O
XA3S200A	200K	4,032	32	16	448	1,792	28K	288K	16	4	195	90
XA3S400A	400K	8,064	40	24	896	3,584	56K	360K	20	4	311	142
XA3S700A	700K	13,248	48	32	1,472	5,888	92K	360K	20	8	372	165
XA3S1400A	1400K	25,344	72	40	2,816	11,264	176K	576K	32	8	375	165

Tabla 21. Resumen de características de la serie SpartanXA-3A

3.4.3. Altera: Serie Cyclone

Cuenta con dos variantes, la Cyclone2 y la Cyclone3. Orientadas a los mercados con alto nivel de producción y muy sensibles a los costes. Todas ellas se fabrican con tecnología CMOS en obleas de 300mm con nivel de integración de 65nm de baja potencia para las Cyclone3 y de 90nm para Cyclone2. Su tecnología de programación está basada en celdas SRAM. La disposición de sus bloques de entrada/salida es en dos filas alrededor del dado para conseguir reducir el tamaño. Las entradas y salidas quedan escalonadas.

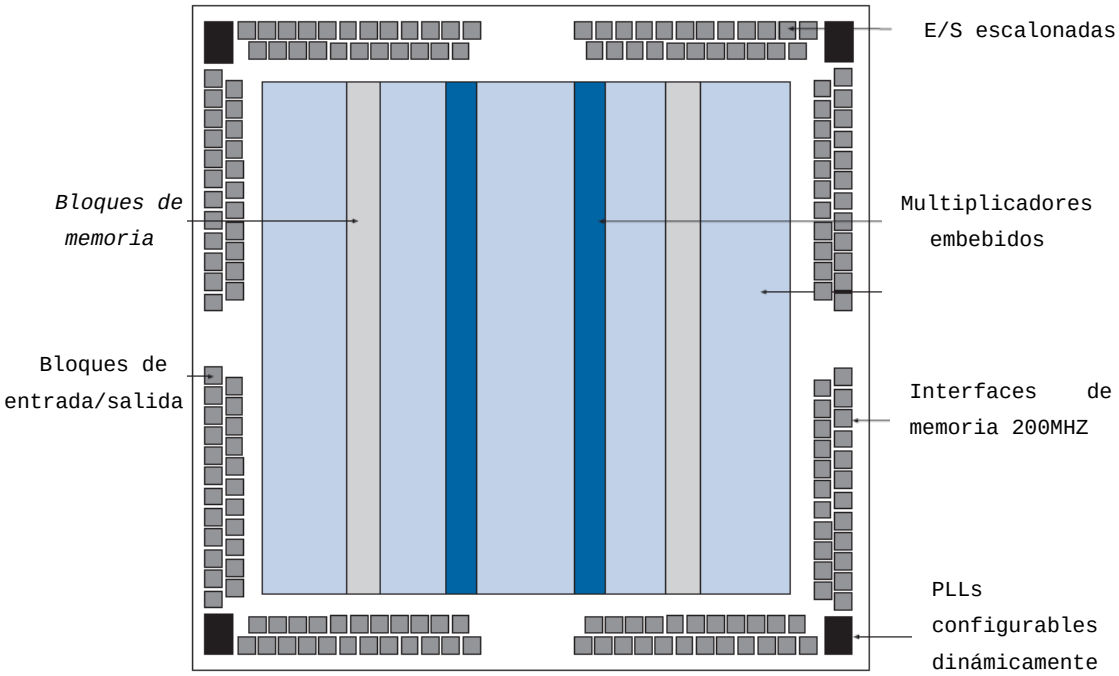


Figura 27. Organización Cyclone III (Fuente: Altera)

3.4.3.1 Arquitectura

Se revisa, de acuerdo a la Figura 27, los principales elementos estructurales que conforman su arquitectura.

Elementos Lógicos: forman una matriz y se denominan LAB (*logic block array)*. Cada uno de estos LAB se componen de 16 unidades formadas por los dos elementos básicos: tabla de búsqueda de cuatro entradas (LUT) y por un registro que puede programarse como de tipo D, T, JK o RS. Cada uno de esos 16 elementos se denomina *Logic Element(LE)* y se considera la unidad lógica mínima. Los LEs cuentan con otros recursos al igual que la celda básica de Xilinx que son una cadena de acarreo que también es utilizada para la conexión de registros.

Cada LE dispone de tres salidas hacia distintos tipos de recursos de conexión: local, de fila o de columna. Cabe destacar la posibilidad de utilizar la LUT y el registro de forma independiente aprovechado las diferentes salidas disponibles. Dispone de líneas específicas para encadenar los registros de los distintos LE ubicados en el mismo LAB. El acarreo de salida de cada LE puede encadenarse con el de entrada del siguiente a lo largo de todo el LAB.

Bloques de entrada/salida: se agrupan por bancos debido a la posibilidad de configurarse individualmente para adaptarse a distintos estándares. Entre sus características se recogen:

- Disponible salidas simples referenciadas o no y salidas diferenciales.
- Control de la corriente de salida para alguno de los estándares
- Pendiente de cambio (*slew-rate)* programable
- Salidas drenador abierto
- Mantiene la señal de una E/S en su último valor mientras que el bus está en triestado (Bus-Hold)
- Incluye una impedancia de salida configurable para evitar rebotes en las líneas de transmisión (*On-*

chip termination, OCT). Elimina la necesidad de resistores externos.

- Resistencia de pull-up programable
- Retardo programable de la E/S
- Diodo de protección

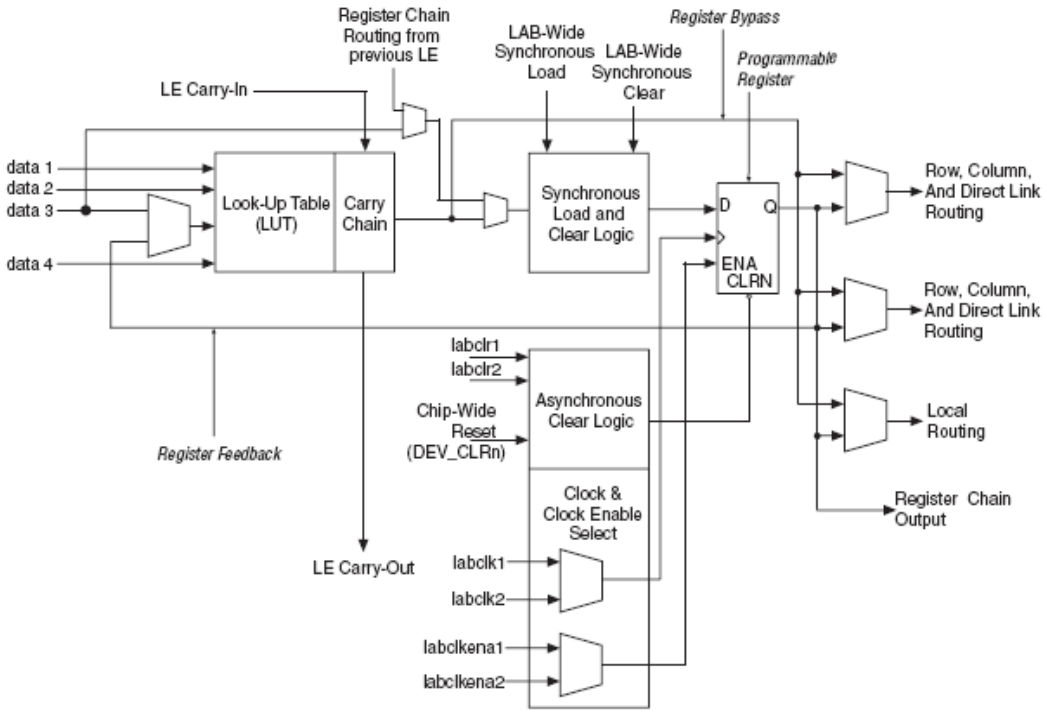


Figura 28. Detalle de un elemento lógico (LE) de la FPGA Cyclone III de Altera

Bloques de memoria: varía el total dependiendo de cada dispositivo. Cada módulo de memoria tiene una capacidad de 8.192 bits más 1.024 destinados al control de paridad. Pueden configurarse para conseguir diferentes organizaciones y disponen de puerto doble para poder realizar dos lecturas, dos escrituras o escritura/lectura simultánea.

Multiplicadores: los bloques multiplicadores pueden utilizarse como uno único de 18x18 bits o bien como una pareja de 9x9 bits cada uno, pueden encadenarse en cascada para conseguir otros

anchos. Están optimizados para conseguir alto rendimiento en funciones propias de DSP.

PLL: disponibles hasta cuatro ubicados en las esquinas del dispositivo. Su misión es la de sincronizar la fase y frecuencia de una señal de reloj externa o interna con otra de referencia. Para el desplazamiento de fase emplea un oscilador controlador de tensión (VCO) en lugar de elementos de retardos.

3.4.4. Serie Cyclone

Esta firma ofrece la serie Cyclone desde hace algunos años y más recientemente presentó los nuevos modelos Cyclone2 y Cyclone3 con sus capacidades lógicas aumentadas, Tabla 22,Tabla 23.

	EP2C5	EP2C8	EP2C15	EP2C20	EP2C35	EP2C50	EP2C70
Elementos Lógicos (LE)	4.608	8.256	14.448	18.752	33.216	50.528	68.410
Memoria (bits)	119.808	165.888	239.616	239.616	483.840	594.432	1.152.000
Multiplicadores	13	18	26	26	35	86	150
PLLs	2	2	4	4	4	4	4
Número máximo de I/O de usuario	158	182	315	315	475	450	622

Tabla 22. Resumen de características de la serie Cyclone2 (Fuente: Altera)

	EP3C5	EP3C10	EP3C16	EP3C25	EP3C40	EP3C55	EP3C80	EP3C120
Elementos Lógicos (LE)	5.136	10.320	15.408	24.624	39.600	55.856	81.264	119.088
Memoria (Kbits)	414	414	504	594	1.134	2.340	2.745	3.888
Multiplicadores	23	23	56	66	126	156	244	288
PLLs	2	2	4	4	4	4	4	4
Redes globales de reloj	10	10	20	20	20	20	20	20

Tabla 23. Resumen de características de la serie Cyclone3 (Fuente: Altera)

3.4.5. ACTEL: Igloo y ProAsicPlus

Este fabricante ofrece FPGAs clasificadas de una forma diferente a los restantes citados. Aunque entre sus dispositivos los hay de bajo coste, optan por destacar otras cualidades como el bajo consumo, integración mixta o la tolerancia a radiaciones. Para el presente estudio se toman los modelos que forman la serie de bajo consumo, al ser también las de menor precio y de mayores similitudes con las de otros fabricantes.

Puestos a diferenciarse, Actel lo hace también ofreciendo FPGAs económicas con tecnología de configuración no volátil basadas en memoria FLASH y densidad de integración de 130nm. Están especialmente diseñadas para ser utilizadas en equipos portátiles dado su extremado bajo consumo y reducido tamaño.

3.4.5.1 Arquitectura

La arquitectura interna se construye a partir de los componentes indicados en la Figura 29 y que se describen brevemente.

Elementos Lógicos:

El elemento básico se denomina VersaTile está basado en multiplexores y puede configurarse de cuatro formas distintas:

- Función lógica de tres entradas (LUT-3)
- Latch con entradas clear y set
- Flip-flop tipo D con entradas clear y set
- Flip-Flop D con entrada enable, clear y set

Dispone de cuatro entradas y de dos salidas y puede interconectarse con otros VersaTiles usando cualquiera de los

cuatro niveles de conexión. Los VersaTiles están organizados matricialmente en la zona central, en la parte superior e inferior del dado se ubican el resto de componentes embebidos y alrededor los bloques de entrada/salida.

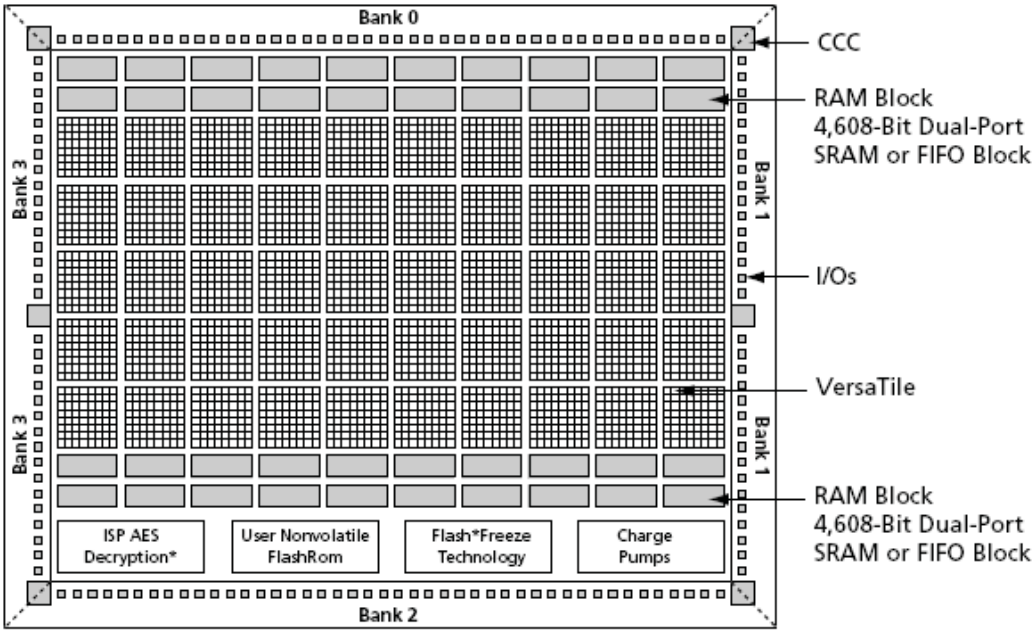


Figura 29. Arquitectura de la serie Igloo (Fuente: www.actel.com)

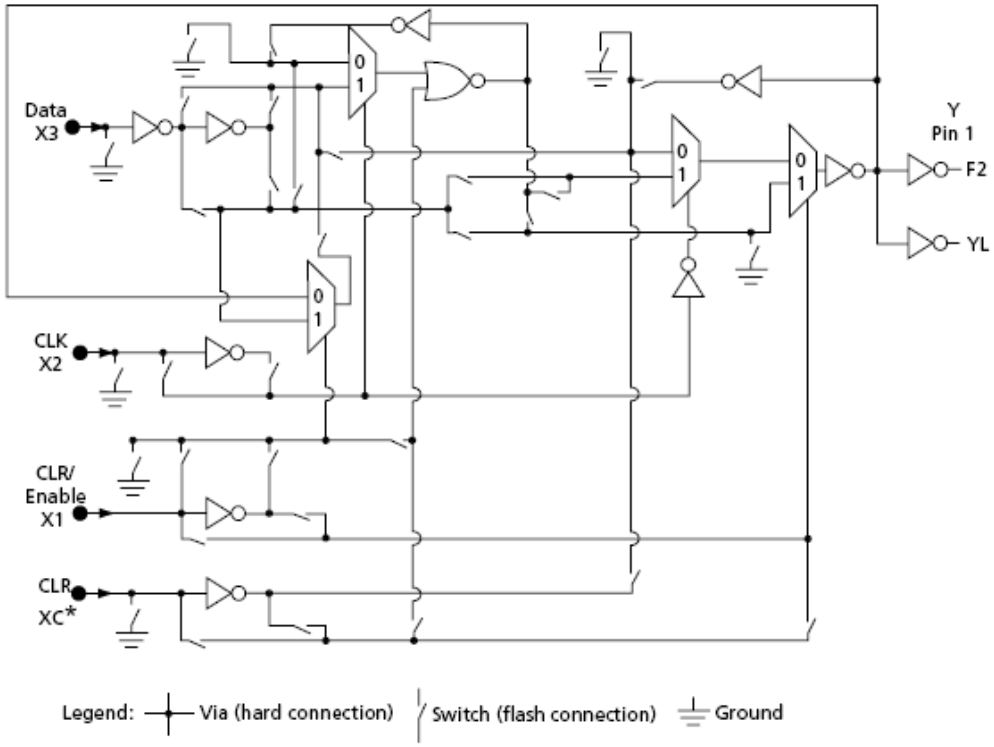


Figura 30. Vista detallada de una celda lógica VersaTile (Fuente: www.actel.com)

Bloque de entrada/salida:

Según dispositivo existen de dos a cuatro bancos de bloques de entrada/salida alrededor del núcleo. Al igual que en las FPGAs de otros fabricantes cada uno de los bloques dispone de un terminal de alimentación independiente para soportar distintos estándares tanto simples como diferenciales. Cuenta con la circuitería necesaria para implementar SDR y DDR.

Cada elemento de entrada y salida puede ser configurado para variar la pendiente de cambio, la corriente de salida, el valor de una resistencia terminadora o de pull-up y pull-down.

Los estándares soportados son:

- Simples
    - LVTTL / LVCMOS (3.3 V)
    - LVCMOS (2.5V, 1.8V, 1.5V)
    - LVCMOS (2.5V y 5V)
    - PCI, PCI-X (3.3V)
- Diferencial:
    - LVPECL
    - LVDS
    - B-LVDS
    - M-LVDS

Bloques de memoria SRAM:

En la parte superior e inferior del dado existen bloques de memoria SRAM con capacidad de 4.608 bits con puertos de lectura y escritura independientes. Cada bloque puede organizarse con distinta anchura y profundidad, desde 4.608 x 1 hasta 256 x 18. Los bloques de RAM pueden ser configurados como memorias FIFO sin necesidad de hacer uso de VersaTiles.

Memoria flash de usuario:

Disponible para ser programada a través del puerto JTAG y no desde el núcleo, puede, entre otras aplicaciones, ser usada para almacenar la clave de desencriptación AES, dando la posibilidad de actualizar la funcionalidad del dispositivo a través de redes públicas de forma segura. Su capacidad es de 1 Kbit.

Circuito acondicionador de reloj (CCC):

Existen seis bloques CCC en los dispositivos Igloo, pueden incluir desde un circuito PLL hasta seis. Establecen

desplazamientos de fase, división/multiplicación de frecuencias, retrasos y realimentación externa. El rango de frecuencias que acepta es de 1,5 MHZ hasta 250 MHz.

3.4.5.2 Series Igloo y ProAsic

- **Serie Igloo/e, Serie Igloo/e M1**:

  Con un bajo consumo ofrece el mayor número de entradas/salidas disponibles para el usuario, y algunos modelos incluyen licencia del procesador soft-core ARM1.

	AGL015	**AGL030**	**AGL060**	**AGL125**	**AGL250**	**AGL400**	**AGL600**	**AGL1000**	**AGLE600**	**AGLE3000**
					MA1GL250	**MA1GL400**	**MA1GL600**	**MA1GL1000**		**MA1GLE3000**
Puertas	15k	30k	60k	125k	250k	400k	600k	1M	600k	3M
Macros	128	256	512	1.024	-	-	-	-	-	-
Versa-Tiles	384	768	1.536	3.072	6.144	9.216	13.824	24.576	13.824	75.264
uW	5	5	10	16	24	32	36	53	49	137
Bloques RAM	-	-	18	36	36	54	108	144	108	504
Flash	1 k	1 k	1 k	1 k	1 k	1 k	1 k	1 k	1 k	1 k
Seguridad AES (no para ARM)	-	-	Yes	Yes	Yes / No	Yes / No	Yes / No	Yes / No	Yes	Yes / No
PLLs	-	-	1	1	1	1	1	1	6	6
Versa-Net globales	6	6	18	18	18	18	18	18	18	18
Bancos de E/S	2	2	2	2	4	4	4	4	8	8
Vel.	-F, Std.	-F, Std.	-F, Std.	-F, Std.	-F, Std.	-F, Std.	-F, Std.	-F, Std.	-F, Std.	-F, Std.
Tª	C, I	C, I	C, I	C, I	C, I	C, I	C, I	C, I	C, I	

Tabla 24. Dispositivos de la serie Igloo e IglooE, incluye también los modelo MA1

A. IglooPlus: cuenta con dispositivos que integran un mayor número de bloque lógicos y prestaciones mejoradas en las entradas y salidas.

IGLOO PLUS	AGLP030	AGLP060	AGLP125
Puertas de sistemas	30 k	60 k	125 k
Macroceldas equivalentes	256	512	1,024
VersaTiles (D-Flip-Flops)	792	1,584	3,12
Flash*Freeze Mode (µW)	5	10	16
RAM kbits (1,024 bits)	-	18	36
Bloques de 4,608	-	4	8
FlashROM (bits)	1 k	1 k	1 k
Seguridad (AES) ISP	-	Yes	Yes
PLLs en CCCs	-	1	1
VersaNet Globales	6	18	18
E/S Estandares	IGLOO PLUS	IGLOO PLUS	IGLOO PLUS
Bancos de E/S	4	4	4
Grados de velocidad	-F, Std.	-F, Std.	-F, Std.
Temperatura	C, I	C, I	C, I
PVP	11,79	14,43	20,71

Tabla 25. Modelos de la serie IglooPlus

B. IglooNano: serie de menor tamaño y menor consumo con un número reducido de entradas y salidas

Igloo Nano	AGLN010	AGLN015	AGLN020	AGLN0301	AGLN060	AGLN125	AGLN250
Puertas de sistemas	10 k	15 k	20 k	30 k	60 k	125 k	250 k
Macroceldas eqivalentes	86	128	172	256	512	1,024	2,048

Igloo Nano	AGLN010	AGLN015	AGLN020	AGLN0301	AGLN060	AGLN125	AGLN250
VersaTiles (D-Flip-Flops)	260	384	520	768	1,536	3,072	6,144
Flash*Freeze Mode (µW)	2	4	4	5	10	16	24
RAM kbits (1,024 bits)	-	-	-	-	18	36	36
Bloques de 4,608	-	-	-	-	4	8	8
FlashROM (bits)	1k	1k	1k	1k	1k	1k	1k
Seguridad (AES) ISP	-	-	-	-	Yes	Yes	Yes
PLLs en CCCs	-	-	-	-	1	1	1
VersaNet Globales	4	4	4	6	18	18	18
E/S Estándares	Std., Hot-Swap	Std., Hot-Swap	Std., Hot-Swap	Std., Hot-Swap	Std., Hot-Swap	Std., Hot-Swap	Std., Hot-Swap
Bancos de E/S	2	3	3	2	2	2	4
Grados de velocidad	Std.	Std.	Std.	Std.	Std.	Std.	Std.
Temperatura	C, I	C, I	C, I	C, I	C, I	C, I	C,I

Tabla 26. Modelos de la serie Igloo Nano

- **Serie ProAsic3:** ofrece modelos que soportan las temperaturas necesarias en aplicaciones de automoción y de tipo militar.

  A. ProAsic3/E: similar a la Igloo/E pero con mejor rendimiento, puede llegar a frecuencias de hasta 350MHz manteniendo un reducido consumo y precio.

ProASIC3 Devices	**A3P015**	**A3P030**	**A3P060**	**A3P125**	**A3P250**	**A3P400**	**A3P600**	**A3P1000**
Core MP7								**M7A3P1000**
Cortex-M1					**M1A3P250**	**M1A3P400**	**M1A3P600**	**M1A3P1000**
Puerta de Sistema	15 k	30 k	60 k	125 k	250 k	400 k	600 k	1 M
Macroceldas equivalentes	128	256	512	1,024	-	-	-	-
VersaTiles (D-Flip-Flop)	384	768	1,536	3,072	6,144	9,216	13,824	24,576
RAM kbits (1,024 bits)	-	-	18	36	36	54	108	144
4,608-Bit Bloques	-	-	4	8	8	12	24	32
FlashROM (bits)	1 k	1 k	1 k	1 k	1 k	1 k	1 k	1 k
Secure (AES) ISP	No	No	Yes	Yes	Yes1	Yes1	Yes1	Yes1
PLLs	-	-	1	1	1	1	1	1
VersaNet Globals	6	6	18	18	18	18	18	18
Estándares de E/S	Std. & Hot Swap	Std. & Hot Swap	Std+	Std+	Std+/LVDS	Std+/LVDS	Std+/LVDS	Std+/LVDS
Bancos de E/S	2	2	2	2	4	4	4	4
Velocidad	-F, Std.	-F, Std.,-1,-2	-F, Std.,-1,-2	-F, Std.,-1,-2	-F, Std.,-1,-2	-F, Std.,-1,-2	-F, Std.,-1,-2	-F, Std.,-1,-2
Temperatura	C, I	C, I	C, I, T	C, I, T	C, I, T	C, I	C, I	C, I, T, M

Tabla 27. Dispositivos de la serie ProAsic3 (Fuente: www.actel.com)

B. ProAsic3L: añade sobre la anterior un mayor número de lógica.

ProASIC3L Family	A3P250L	A3P600L	A3P1000L	A3PE600L	A3PE3000L
ARM Cortex-M1-Enabled		M1A3P600L	M1A3P1000L		M1A3PE3000L
Puerta de Sistema	250 k	600 k	1 M	600 k	3 M
VersaTiles (D-Flip-Flop)	6,144	13,824	24,576	13,824	75,264
RAM kbits (1,024 bits)	36	108	144	108	504
4,608-Bit Bloques	8	24	32	24	112
FlashROM (bits)	1 k	1 k	1 k	1 k	1 k
Secure (AES) ISP	Yes	Yes	Yes	Yes	Yes
PLLs	1	1	1	6	6
VersaNet Globals	18	18	18	18	18
Estándars de E/S	Std.+/LVDS	Std.+/LVDS	Std.+/LVDS	Pro	Pro
Bancos de E/S	4	4	4	8	8
Flash*Freeze Power (mW) a VCC=1.2 V	0.33	0.66	1.06	TBA	3.30
Velocidad	Std., -1	Std., -1	Std., -1	Std., -1	Std., -1
Temperatura	C, I	C, I	C, I	M	C, I, M

Tabla 28. Dispositivos de la serie ProAsic3L (Fuente: www.actel.com)

C. ProAsic3Nano: aunque son las de menor tamaño de esta familia mantienen un alto rendimiento, reducido consumo y mínimo precio. A cambio, el número de entradas/salidas de usuario disminuye.

ProASIC3 nano	A3PN010	A3PN015	A3PN020	A3PN0301	A3PN060	A3PN125	A3PN250
Puerta de Sistema	10 k	15 k	20 k	30 k	60 k	125 k	250 k
Macroceldas equivalentes	86	128	172	256	512	1,024	2,048
VersaTiles (D-Flip-Flop)	260	384	520	768	1,536	3,072	6,144

ProASIC3 nano	A3PN010	A3PN015	A3PN020	A3PN0301	A3PN060	A3PN125	A3PN250
RAM kbits (1,024 bits)	-	-	-	-	18	36	36
4,608-Bit Bloques	-	-	-	-	4	8	8
FlashROM (bits)	1 k	1 k	1 k	1 k	1 k	1 k	1 k
Secure (AES) ISP	-	-	-	-	Yes	Yes	Yes
PLLs	-	-	-	-	1	1	1
VersaNet Globals	4	4	4	6	18	18	18
Estándares de E/S	Std., Hot Swap	Std., Hot Swap	Std., Hot Swap	Std., Hot Swap	Std., Hot Swap	Std., Hot Swap	Std., Hot Swap
Bancos de E/S	2	3	3	2	2	2	4
Velocidad	Std., -1, -2	Std., -1, -2	Std., -1, -2	Std., -1, -2	Std., -1, -2	Std., -1, -2	Std., -1, -2
Temperatura	C, I	C, I	C, I	C, I	C, I	C, I	C, I

Tabla 29. Dispositivos de la serie ProAsic3 Nano (Fuente: www.actel.com)

3.4.6. LATTICE SEMICONDUCTORS: ECP2/M Y XP2

Este fabricante ofrece dos familias de FPGAs de bajo coste fabricado con tecnología CMOS y nivel de integración de 90nm, reciben el nombre de ECP2 (*Economy Plus)* y XP. La primera familia emplea configuración basada en memoria SRAM y la segunda en FLASH. Su arquitectura incluye una serie de elementos comunes y otros característicos de cada modelo.

3.4.6.1 Arquitectura

De acuerdo a la arquitectura mostrada en la Figura 31 , sus elementos estructurales son:

Elementos Lógicos: Los bloque lógicos están organizados matricialmente y salpicados entre ellos hay filas de bloques de memoria y de procesamiento digital de señal. Alrededor de este entramado se disponen los bloques de entrada y salida.

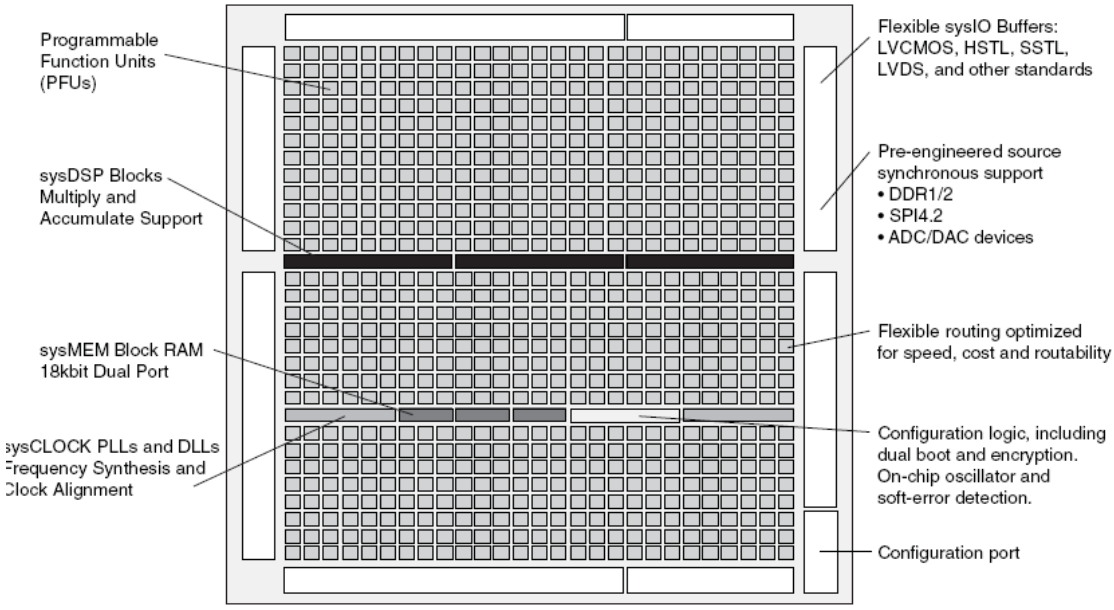


Figura 31. Diagrama de bloques del dispositivo ECP2-6

El bloque lógico de mayor jerarquía en estas FPGAs se denomina PFU y está formado por cuatro *slices* conectados entre sí, cada uno de éstos puede considerarse la mínima unidad lógica. Tres de ellos son idénticos y constan de dos tablas de búsqueda de 4 entradas (LUT) cuyas salidas se dirigen hacia un registro, el cuarto *slice* difiere en que no incluye este elemento. Cada PFU dispone además de lógica adicional de interconexión o propia de tareas aritméticas. Cada *slice* puede configurarse para actuar de cuatro modos diferentes:

- Lógico: cada LUT de cada *slice* se configura como tabla combinacional de 4 entradas. Uniendo ambas LUT del *slice* se obtiene una tabla de 5 entradas.
- Aritmético: En cada *slice* se pueden implementar pequeñas funciones aritméticas como sumas o restas de dos bits entre otras. En este modo se puede hacer uso de las líneas de acarreo dedicadas.
- RAM: las LUT se emplean para formar memoria RAM distribuida de dos tipos: memoria de un solo puerto de 16x4 bits (SPR) ó memoria de pseudo-doble puerto de 16x2 bits (PDPR).
- ROM: las LUT se configuran como memoria ROM que reciben su información durante la configuración del PFU.

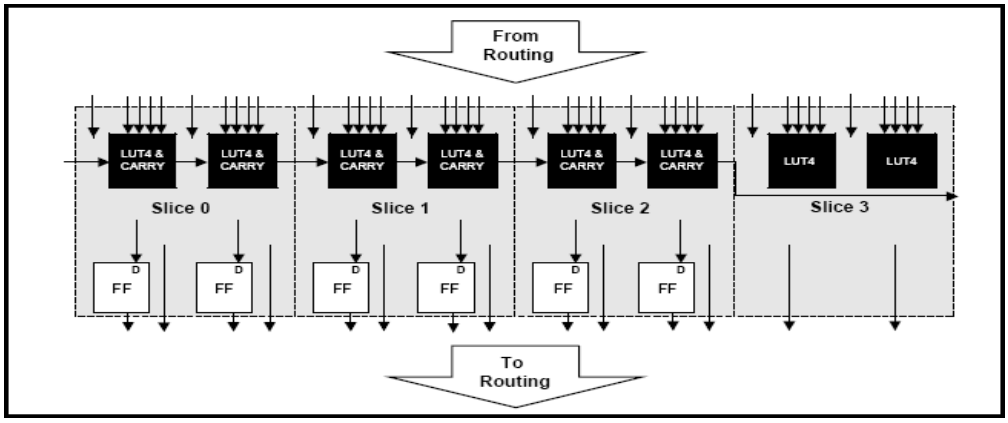


Figura 32. Detalle de la unidad PFU (Fuente: Lattice Semiconductor)

Bloques de entrada/salida: son celdas denominadas PIC (*programable I/O cell)* que incluyen dos elementos de entrada/salida (PIO-A y PIO-B). Cada PIO dispone de su propio buffer y puede actuar como entrada o salida, disponiendo ambos de control triestado. Ambos se combinan para dar lugar a entradas/salidas diferenciales.

Estándares soportados:

- Simples
  - LVCMOS / LVTTL (840 Mbps) con las siguientes capacidades:
    - ✓ Pendiente de cambio (slew rate) programable
    - ✓ Corriente de salida programable
    - ✓ Resistencias de pull-up y pull-down programables
    - ✓ Posibilidad de configuración de salida drenador abierto.
  - PCI, LVDS, SSTL, HSTL
- Diferenciales: LVPECL, BLVDS, RSDS , MLVDS

Disponen de circuitería para ofrecer interfaz DDR 1 y 2 de hasta 533 MHz y SPI 4.2 a 750 MHz.

Bloques de memoria (EBR): son bloques de memoria de 18 kbits cada uno, el total de bloques depende del dispositivo. Esta memoria puede configurarse con distintas organizaciones desde 16.384x1 hasta 512 x 36 con un único puerto y hasta 1024x18 en doble puerto.

Bloques DSP: existen desde tres hasta 42 bloques según modelo. Cada bloque puede incluir cuatro tipo de elementos funcionales y un ancho del camino de datos de 9, 18 o 36. Los elementos funcionales disponibles son:

- MULT (multiplicador)
- MAC (multiplicador, acumulador)
- MULTADDSUB (multiplicador, sumador/restador)
- MULTADDSUBSUM (multiplicador, sumador/restador, acumulador)

El número de elementos disponibles en cada bloque depende del ancho seleccionado y algunos pueden concatenarse para conseguir una alta paralelización en la familia ECP2/M. Cada bloque DSP puede trabajar a 375 MHz dando al dispositivo una productividad total de 54,6 MMACs.

PLL, DLLs: disponen de hasta ocho PLL en dos variedades, dos con circuitería completa denominados GPLL *(General Phase Locked Loop)* y seis que no disponen de circuitería para el ajuste de retrasos y que reciben el nombre de SPLL (*Standard Phase Locked Loop).* Algunos dispositivos de esta familia incluyen también dos bloques DLL.

- **LatticeECP2**:

Dispositivo	ECP2-6	ECP2-12	ECP2-20	ECP2-35	ECP2-50	ECP2-70
LUTs (K)	6	12	21	32	48	68
RAM distribuida (Kbits)	12	24	42	64	96	136
EBR SRAM (Kbits)	55	221	276	332	387	1032
EBR SRAM Bloques	3	12	15	18	21	60
sysDSP Blocks	3	6	7	8	18	22
18x18 Multiplicadores	12	24	28	32	72	88
GPLL + SPLL + DLL	2+0+2	2+0+2	2+0+2	2+0+2	2+2+2	2+4+2
Máximo de I/O	190	297	402	450	500	583

Tabla 30. Resumen de características de la serie ECP2 (Fuente: Lattice Semiconductors)

3.4.6.2 Serie Lattice

Se hace referencia a las siguientes subseries:

- **LatticeECP2M**: ésta serie incluye bloques embebidos destinados a las comunicaciones de alta velocidad. Los bloques SerDes incluyen circuitería para conversión serie/paralelo–paralelo/serie (*Serializar/Deserializer - SerDes*). En total cuenta con hasta 16 canales agrupados de cuatro en cuatro (por esto reciben el nombre de Quad) en cada una de las cuatro esquinas (cantidad según modelo). Están orientados a la implementación de protocolos basados en 8b/10b tales como Gigabit-Ethernet o PCI-Express entre otros. Cada canal contiene circuitería de recepción/transmisión para transferir datos serie de alta velocidad pudiendo llegar a tasas de hasta 3,125 Gbps.

Dispositivo	ECP2M20	ECP2M35	ECP2M50	ECP2M70	ECP2M100
LUTs (K)	19	34	48	67	95
RAM distribuida (Kbits)	66	114	225	246	288
EBR SRAM (Kbits)	1217	2101	4147	4534	5308
EBR SRAM Bloques	41	71	101	145	202
Bloques sysDSP	6	8	22	24	42
18x18 Multiplicadores	24	32	88	96	168
GPLL + SPLL + DLL	2+6+2	2+6+2	2+6+2	2+6+2	2+6+2
Máximo de I/O	304	410	410	436	520
Bloques sysSERDES (Quad)	1	1	2	4	4

Tabla 31. Resumen de características de la serie ECP2M (Fuente: Lattice Semiconductors)

- **LatticeXP2**: incluye memoria de tipo FLASH para almacenar la configuración del dispositivo. Cuando se conecta la alimentación se transfiere el contenido de las FLASH hacia la SRAM de configuración. Al integrar memoria no volátil se ahorra la necesidad de añadir un circuito externo para

guardar el fichero de configuración, aportando una mayor seguridad del diseño. La memoria FLASH está ubicada a la derecha y la izquierda de la matriz de bloques lógicos. Puede ser utilizadas para almacenar datos de usuario que no quieran perderse al desconectar la alimentación.

Dispositivo	**XP2-5**	**XP2-8**	**XP2-17**	**XP2-30**	**XP2-40**
LUTs (K)	5	8	17	29	40
RAM distribuida (Kbits)	10	18	35	56	83
EBR SRAM (Kbits)	166	221	276	387	885
EBR SRAM Bloques	9	12	15	21	48
sysDSP Blocks	3	4	5	7	8
18x18 Multiplicadores	12	16	20	28	32
TENSIÓN VCC	1.2	1.2	1.2	1.2	1.2
GPLL	2	2	4	4	4
Máximo de I/O	172	201	358	472	540

Tabla 32. Resumen de características de la serieXP2 (Fuente: Lattice Semiconductors)

3.5. Selección de FPGAs

Tras la revisión efectuada sobre fabricantes y FPGAs disponibles, toca decidir el subconjunto de las mismas que se incluirán como tecnologías objetivos del PdC que se determine.

Según los apartados anteriores se concluye que las denominadas de bajo coste han sido la revelación en los últimos años, todos los fabricantes mejoran sus series pertenecientes a este grupo y las adaptan a los nuevos mercados como el de la automoción. Los usuarios por su parte, encuentran en ellas una opción económica y flexible para sus diseños más dinámicos y menos longevos. Las más antiguas no ofrecían demasiados recursos ni eran demasiado potentes, esto causó que casi todos los PdC planteados estuviesen dirigidos a las series de alto rendimiento. Este resurgir de las FPGAs más económicas unido a la proliferación de nuevos modelos, a veces muy similares, sienta las bases para enfocarlas como objetivo de un nuevo patrón de pruebas que permita clasificarlas.

Dado el vasto conjunto de FPGAs disponibles, se establece como criterio de filtrado para esta tesis el de un precio por unidad, en concreto se considerarán aquellos modelos que se oferten por 50$ o menos.

Por otro lado se opta por trabajar solo con los fabricantes Xilinx Inc. y Altera Co. En primer lugar debido a su situación dominante en el mercado, lo que hace de sus productos un objetivo fundamental de puntuación y clasificación. Y en segundo lugar, debido al hecho de que es posible disponer de una versión gratuita de sus entornos de trabajo sin limitación temporal, con soporte para las tecnologías seleccionadas y ofreciendo todas las herramientas necesarias para completar las etapas de diseño necesarias para realizar el PdC.

A continuación se ofrecen las tablas que recogen los modelos de FPGAs, Spartan3, CycloneII y CycloneIII que se han seleccionado.

Modelo	Precio	Encapsulado	Puertas	LUTs_FF	Fint_MHz	V	I/O	RAM_bits	Flash_bits	Mult_18x18	Ges_clk	T_MAX	T_MIN
EP2C5T144C8	12,80	144TQFP	5000	4608	500,00	1,15	89	119808	0	13	2	85	0
EP3C5E144C8	12,80	144-EQFP	5000	5136	500,00	1,14	94	423936	0	23	2	85	0
EP2C5Q208C8	13,90	208PQFP	5000	4608	500,00	1,15	142	119808	0	13	2	85	0
EP2C5F256C8	14,90	256-FBGA	5000	4608	500,00	1,15	158	119808	0	13	2	85	0
EP3C5F256C8	14,90	256-FBGA	5000	5136	500,00	1,14	182	423936	0	23	2	85	0
EP2C5T144C7	15,30	144TQFP	5000	4608	450,00	1,15	89	119808	0	13	2	85	0
EP2C5T144I8	15,30	144TQFP	5000	4608	500,00	1,15	89	119808	0	13	2	100	-40
EP3C5E144C7	15,30	144-EQFP	5000	5136	450,00	1,14	94	423936	0	23	2	85	0
EP3C5U256C8	16,50	256-UBGA	5000	5136	500,00	1,14	182	423936	0	23	2	85	0
EP2C5Q208C7	16,70	208PQFP	5000	4608	450,00	1,15	142	119808	0	13	2	85	0
EP2C5Q208I8	16,70	208PQFP	5000	4608	500,00	1,15	142	119808	0	13	2	100	-40
EP2C5F256C7	17,90	256-FBGA	5000	4608	450,00	1,15	158	119808	0	13	2	85	0
EP2C5F256I8	17,90	256-FBGA	5000	4608	500,00	1,15	158	119808	0	13	2	100	-40
EP3C5F256C7	17,90	256-FBGA	5000	5136	450,00	1,14	182	423936	0	23	2	85	0
EP3C5M164C7	17,90	164-MBGA	5000	5136	450,00	1,14	106	423936	0	23	2	85	0
EP2C5T144C6	19,20	144TQFP	5000	4608	402,50	1,15	89	119808	0	13	2	85	0
EP2C8T144C8	19,20	144TQFP	8000	8256	500,00	1,15	85	165888	0	18	2	85	0
EP3C10E144C8	19,20	144-EQFP	10000	10320	500,00	1,14	94	423936	0	23	2	85	0
EP3C5E144I7	19,20	144-EQFP	5000	5136	450,00	1,14	94	423936	0	23	2	100	-40
EP3C5U256C7	19,70	256-UBGA	5000	5136	450,00	1,14	182	423936	0	23	2	85	0
EP2C8Q208C8	20,30	208PQFP	8000	8256	500,00	1,15	138	165888	0	18	2	85	0
EP2C5F256C6	22,40	256-FBGA	5000	4608	402,50	1,15	158	119808	0	13	2	85	0
EP2C8T144C7	23,10	144TQFP	8000	8256	450,00	1,15	85	165888	0	18	2	85	0
EP2C8T144I8	23,10	144TQFP	8000	8256	500,00	1,15	85	165888	0	18	2	100	-40
EP3C10E144C7	23,10	144-EQFP	10000	10320	450,00	1,14	94	423936	0	23	2	85	0
EP2C8F256C8	23,50	256-FBGA	8000	8256	500,00	1,15	182	165888	0	18	2	85	0
EP3C10F256C8	23,50	256-FBGA	10000	10320	500,00	1,14	182	423936	0	23	2	85	0

Modelo	Precio	Encapsulado	Puertas	LUTs_FF	Fint_MHz	V	I/O	RAM_bits	Flash_bits	Mult_18x18	Ges_clk	T_MAX	T_MIN
EP2C8Q208C7	24,30	208PQFP	8000	8256	450,00	1,15	138	165888	0	18	2	85	0
EP2C8Q208I8	24,30	208PQFP	8000	8256	500,00	1,15	138	165888	0	18	2	100	-40
EP3C10U256C8	25,80	256-UBGA	10000	10320	500,00	1,14	182	423936	0	23	2	85	0
EP3C5E144A7	26,40	144-EQFP	5000	5136	450,00	1,20	94	423936	0	23	2	125	-40
EP3C16E144C8	26,70	144-EQFP	16000	15408	500,00	1,14	84	516096	0	56	4	85	0
EP3C5F256C6	26,80	256-FBGA	5000	5136	402,50	1,14	182	423936	0	23	2	85	0
EP3C5F256I7	26,80	256-FBGA	5000	5136	450,00	1,14	182	423936	0	23	2	100	-40
EP2C8AF256I8	28,10	256-FBGA	8000	8256	500,00	1,15	182	165888	0	18	2	100	-40
EP2C8F256C7	28,10	256-FBGA	8000	8320	450,00	1,15	182	165888	0	18	2	85	0
EP2C8F256I8	28,10	256-FBGA	8000	8256	500,00	1,15	182	165888	0	18	2	100	-40
EP3C10F256C7	28,10	256-FBGA	10000	10320	450,00	1,14	182	423936	0	23	2	85	0
EP3C16M164C8	28,50	164-MBGA	16000	15408	500,00	1,14	92	516096	0	56	4	85	0
EP2C8T144C6	28,80	144TQFP	8000	8256	402,50	1,15	85	165888	0	18	2	85	0
EP3C10E144I7	28,80	144-EQFP	10000	10320	450,00	1,14	94	423936	0	23	2	100	-40
EP3C16Q240C8	29,10	240-PQFP	16000	15408	500,00	1,14	160	516096	0	56	4	85	0
EP3C16F256C8	29,60	256-FBGA	16000	15408	500,00	1,14	168	516096	0	56	4	85	0
EP3C5U256C6	29,60	256-UBGA	5000	5136	402,50	1,14	182	423936	0	23	2	85	0
EP3C5U256I7	29,60	256-UBGA	5000	5136	450,00	1,14	182	423936	0	23	2	100	-40
EP3C10U256C7	30,90	256-UBGA	10000	10320	450,00	1,14	182	423936	0	23	2	85	0
EP3C16U256C8	31,70	256-UBGA	16000	15408	500,00	1,14	168	516096	0	56	4	85	0
EP3C16E144C7	32,00	144-EQFP	16000	15408	450,00	1,14	84	516096	0	56	4	85	0
EP3C16F484C8	33,20	484FBGA	16000	15408	500,00	1,14	346	516096	0	56	4	85	0
EP3C16M164C7	34,30	164-MBGA	16000	15408	450,00	1,14	92	516096	0	56	4	85	0
EP2C15AF256C8	34,70	256-FBGA	15000	14448	500,00	1,15	158	239616	0	26	4	85	0
EP2C8F256C6	35,20	256-FBGA	8000	8256	402,50	1,15	182	165888	0	18	2	85	0
EP3C10F256C6	35,20	256-FBGA	10000	10320	402,50	1,14	182	423936	0	23	2	85	0
EP3C10F256I7	35,20	256-FBGA	10000	10320	450,00	1,14	182	423936	0	23	2	100	-40

Modelo	Precio	Encapsulado	Puertas	LUTs_FF	Fint_MHz	V	I/O	RAM_bits	Flash_bits	Mult_18x18	Ges_clk	T_MAX	T_MIN
EP3C16F256C7	35,50	256-FBGA	16000	15408	450,00	1,14	168	516096	0	56	4	85	0
EP3C16U484C8	36,50	484UBGA	16000	15408	500,00	1,14	346	516096	0	56	4	85	0
EP3C16U256C7	38,00	256-UBGA	16000	15408	450,00	1,14	168	516096	0	56	4	85	0
EP3C10U256C6	38,70	256-UBGA	10000	10320	402,50	1,14	182	423936	0	23	2	85	0
EP3C10U256I7	38,70	256-UBGA	10000	10320	450,00	1,14	182	423936	0	23	2	100	-40
EP2C15AF256C6	38,90	256-FBGA	15000	14448	402,50	1,15	152	239616	0	26	4	85	0
EP2C15AF484C8	38,90	484FBGA	15000	14448	500,00	1,15	315	239616	0	26	4	85	0
EP3C10E144A7	39,50	144-EQFP	10000	10320	450,00	1,20	94	423936	0	23	2	125	-40
EP3C25E144C8	39,50	144-EQFP	25000	24624	500,00	1,14	82	608256	0	66	4	85	0
EP3C16F484C7	39,90	484FBGA	16000	15408	450,00	1,14	346	516096	0	56	4	85	0
EP2C15AF256C7	41,60	256-FBGA	15000	14448	450,00	1,15	158	239616	0	26	4	85	0
EP2C15AF256I8	41,60	256-FBGA	15000	14448	500,00	1,15	152	239616	0	26	4	100	-40
EP2C20F256C8	42,70	256-FBGA	20000	18752	500,00	1,15	152	239616	0	26	4	85	0
EP2C20Q240C8	42,70	240-PQFP	20000	18752	500,00	1,15	142	239616	0	26	4	85	0
EP3C25Q240C8	42,70	240-PQFP	25000	24624	500,00	1,14	148	608256	0	66	4	85	0
EP3C16U484C7	43,90	484UBGA	16000	15408	450,00	1,14	346	516096	0	56	4	85	0
EP3C16F256C6	44,40	256-FBGA	16000	15408	402,50	1,14	168	516096	0	56	4	85	0
EP3C25F256C8	44,80	256-FBGA	25000	24624	500,00	1,14	156	608256	0	66	4	85	0
EP2C15AF484C7	46,70	484FBGA	15000	14448	450,00	1,15	315	239616	0	26	4	85	0
EP2C15AF484I8	46,70	484FBGA	15000	14448	500,00	1,15	315	239616	0	26	4	100	-40
EP2C20F484C8	46,90	484FBGA	20000	18752	500,00	1,15	315	239616	0	26	4	85	0
EP3C25U256C8	46,90	256-UBGA	25000	24624	500,00	1,14	156	608256	0	66	4	85	0
EP3C16U256C6	47,50	256-UBGA	16000	15408	402,50	1,14	156	516096	0	56	4	85	0
EP3C10F256A7	48,30	256-BGA	10000	10320	450,00	1,20	182	423936	0	23	2	125	-40
EP3C25F324C8	49,30	324-FBGA	25000	24624	500,00	1,14	215	608256	0	66	4	85	0
EP3C16F484C6	49,90	484FBGA	16000	15408	402,50	1,14	346	516096	0	56	4	85	0

Tabla 33. Listado de FPGAs de las series CycloneII y III de Altera con precio inferior a 50$

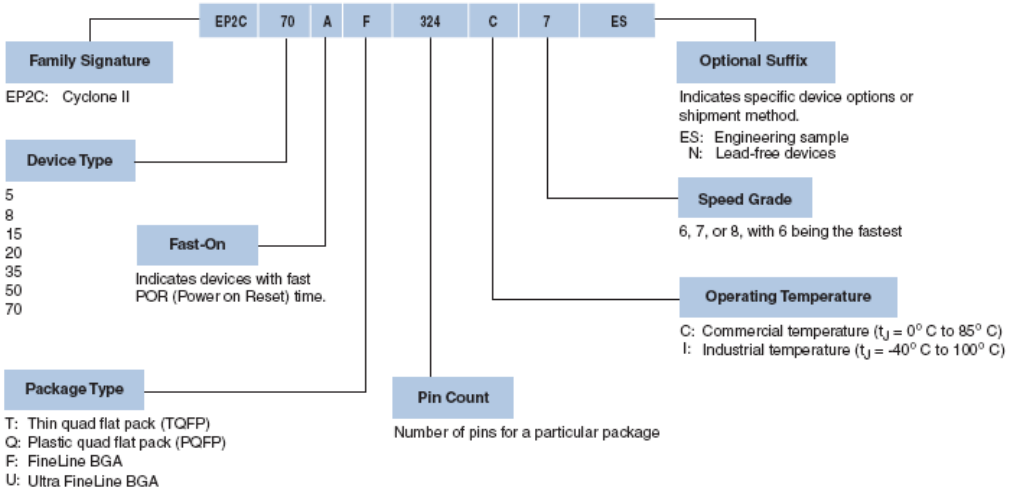


Figura 33. Nomenclatura utilizada por Altera para identificar la serie Cyclone2 (Similar para Cyclone3)

Modelo	Precio	Encapsulado	Puertas	LUTs_FF	Fint_MHz	V	I/O	RAM_bits	Flash_bits	Mult_18x18	Ges_clk	T_MAX	T_MIN
XC3S50A-4VQ100C	5,52	100VTQFP	50.000	1.408	667,00	1,20	68	55.296	0	3	2	85	0
XC3S50A-4VQ100I	6,36	100VTQFP	50.000	1.408	667,00	1,20	68	55.296	0	3	2	100	-40
XC3S50A-5VQ100C	6,60	100VTQFP	50.000	1.408	770,00	1,20	68	55.296	0	3	2	85	0
XC3S50A-4TQ144C	7,44	144TQFP EP	50.000	1.408	667,00	1,20	108	55.296	0	3	2	85	0
XC3S50A-4TQ144I	8,58	144TQFP EP	50.000	1.408	667,00	1,20	108	55.296	0	3	2	100	-40
XC3S50A-4FT256C	8,76	256FTBGA	50.000	1.408	667,00	1,20	144	55.296	0	3	2	85	0
XC3S50A-5TQ144C	8,94	144TQFP EP	50.000	1.408	770,00	1,20	108	55.296	0	3	2	85	0
XC3S50AN-4TQG144C	8,94	144TQFP EP	50.000	1.408	667,00	1,20	108	55.296	1	3	2	85	0
XC3S50-4CP132C	9,30	132CSBGA	50.000	1.536	630,00	1,20	89	73.728	0	50	2	85	0
XC3S50-4TQ144C	9,30	144TQFP	50.000	1.536	630,00	1,20	97	73.728	0	50	2	85	0
XC3S100E-4VQ100C	9,48	100VTQFP	100.000	1.920	572,00	1,20	66	73.728	0	4	2	85	0
XC3S50A-4FT256I	10,08	256FTBGA	50.000	1.408	667,00	1,20	144	55.296	0	3	2	100	-40
XC3S50-4VQ100C	10,10	100VTQFP	50.000	1.536	630,00	1,20	63	73.728	0	50	2	85	0
XC3S50-4VQ100I	10,10	100VTQFP	50.000	1.536	630,00	1,20	63	73.728	0	50	2	100	-40
XC3S50-5VQ100C	10,10	100VTQFP	50.000	1.536	725,00	1,20	63	73.728	0	50	2	85	0
XC3S50AN-4TQG144I	10,32	144TQFP EP	50.000	1.408	667,00	1,20	108	55.296	1	3	2	100	-40
XC3S50A-5FT256C	10,56	256FTBGA	50.000	1.408	770,00	1,20	144	55.296	0	3	2	85	0
XC3S50-4CP132I	10,70	132CSBGA	50.000	1.536	630,00	1,20	89	73.728	0	50	2	100	-40
XC3S50-5CP132C	10,70	132CSBGA	50.000	1.536	725,00	1,20	89	73.728	0	50	2	85	0
XC3S50-4TQ144I	10,70	144TQFP	50.000	1.536	630,00	1,20	97	73.728	0	50	2	100	-40
XC3S50-5TQ144C	10,70	144TQFP	50.000	1.536	725,00	1,20	97	73.728	0	50	2	85	0
XC3S50AN-5TQG144C	10,74	144TQFP EP	50.000	1.408	770,00	1,20	108	55.296	1	3	2	85	0
XC3S100E-4VQ100I	10,92	100VTQFP	100.000	1.920	572,00	1,20	66	73.728	0	4	2	100	-40
XC3S100E-5VQ100C	10,92	100VTQFP	100.000	1.920	657,00	1,20	66	73.728	0	4	2	85	0
XC3S100E-4CP132C	11,10	132CSBGA	100.000	1.920	572,00	1,20	83	73.728	0	4	2	85	0
XC3S100E-4CP132I	11,10	132CSBGA	100.000	1.920	572,00	1,20	83	73.728	0	4	2	100	-40
XC3S200A-4VQ100C	11,60	100VTQFP	200.000	3.584	667,00	1,20	68	294.912	0	16	4	85	0

Modelo	Precio	Encapsulado	Puertas	LUTs_FF	Fint_MHz	V	I/O	RAM_bits	Flash_bits	Mult_18x18	Ges_clk	T_MAX	T_MIN
XC3S200-4VQ100C	11,70	100VTQFP	200.000	3.840	630,00	1,20	63	221.184	0	12	4	85	0
XA3S100E-4VQG100I	12,06	100VTQFP	100.000	1.920	572,00	1,20	66	73.728	0	4	2	100	-40
XC3S50-4PQ208C	12,10	208PQFP	50.000	1.536	630,00	1,20	124	73.728	0	50	2	85	0
XC3S100E-4TQ144C	12,78	144TQFP EP	100.000	1.920	572,00	1,20	108	73.728	0	4	2	85	0
XC3S100E-4TQ144I	12,78	144TQFP EP	100.000	1.920	572,00	1,20	108	73.728	0	4	2	100	-40
XC3S100E-5CP132C	12,78	132CSBGA	100.000	1.920	657,00	1,20	83	73.728	0	4	2	85	0
XC3S100E-5TQ144C	12,78	144TQFP EP	100.000	1.920	657,00	1,20	108	73.728	0	4	2	85	0
XC3S200A-4VQ100I	12,84	100VTQFP	200.000	3.584	667,00	1,20	68	294.912	0	16	4	100	-40
XC3S250E-4VQ100C	13,08	100VTQFP	250.000	4.896	572,00	1,20	66	221.184	0	12	4	85	0
XC3S200A-4FT256C	13,44	256FTBGA	200.000	3.584	667,00	1,20	195	294.912	0	16	4	85	0
XC3S200A-5VQ100C	13,44	100VTQFP	200.000	3.584	770,00	1,20	68	294.912	0	16	4	85	0
XC3S200-4VQ100I	13,45	100VTQFP	200.000	3.840	630,00	1,20	63	221.184	0	12	4	100	-40
XC3S200-5VQ100C	13,45	100VTQFP	200.000	3.840	725,00	1,20	63	221.184	0	12	4	85	0
XA3S100E-4VQG100Q	13,86	100VTQFP	100.000	1.920	572,00	1,20	66	73.728	0	4	2	125	-40
XC3S50-4PQ208I	13,90	208PQFP	50.000	1.536	630,00	1,20	124	73.728	0	50	2	100	-40
XC3S50-5PQ208C	13,90	208PQFP	50.000	1.536	725,00	1,20	124	73.728	0	50	2	85	0
XA3S100E-4CPG132I	14,10	132CSBGA	100.000	1.920	570,00	1,20	83	73.728	0	4	2	100	-40
XA3S100E-4TQG144I	14,10	144TQFP EP	100.000	1.920	572,00	1,20	108	73.728	0	4	2	100	-40
XC3S250E-4VQ100I	15,06	100VTQFP	250.000	4.896	572,00	1,20	66	221.184	0	12	4	100	-40
XC3S250E-5VQ100C	15,06	100VTQFP	250.000	4.896	657,00	1,20	66	221.184	0	12	4	85	0
XC3S200-4TQ144C	15,10	144TQFP	200.000	3.840	630,00	1,20	97	221.184	0	12	4	85	0
XC3S200-4TQ144I	15,10	144TQFP	200.000	3.840	630,00	1,20	97	221.184	0	12	4	100	-40
XC3S200-5TQ144C	15,10	144TQFP	200.000	3.840	725,00	1,20	97	221.184	0	12	4	85	0
XC3S250E-4CP132C	15,30	132CSBGA	250.000	4.896	572,00	1,20	92	221.184	0	12	4	85	0
XC3S250E-4TQ144C	15,30	144TQFP EP	250.000	4.896	572,00	1,20	108	221.184	0	12	4	85	0
XC3S200A-4FT256I	15,42	256FTBGA	200.000	3.584	667,00	1,20	195	294.912	0	16	4	100	-40
XC3S200A-4FG320C	15,78	320FBGA	200.000	3.584	667,00	1,20	248	294.912	0	16	4	85	0

Modelo	Precio	Encapsulado	Puertas	LUTs_FF	Fint_MHz	V	I/O	RAM_bits	Flash_bits	Mult_18x18	Ges_clk	T_MAX	T_MIN
XC3S200AN-4FTG256C	15,84	256FTBGA	200.000	3.584	667,00	1,20	195	294.912	4	16	4	85	0
XC3S400A-4FT256C	15,84	256FTBGA	400.000	7.168	667,00	1,20	195	368.640	0	20	4	85	0
XC3S200A-5FT256C	16,14	256FTBGA	200.000	3.584	770,00	1,20	195	294.912	0	16	4	85	0
XA3S100E-4CPG132Q	16,26	132CSBGA	100.000	1.920	571,00	1,20	83	73.728	0	4	2	125	-40
XA3S100E-4TQG144Q	16,26	144TQFP EP	100.000	1.920	572,00	1,20	108	73.728	0	4	2	125	-40
XA3S250E-4VQG100I	16,68	100VTQFP	250.000	4.896	573,00	1,20	66	221.184	0	12	4	100	-40
XC3S250E-4CP132I	17,58	132CSBGA	250.000	4.896	572,00	1,20	92	221.184	0	12	4	100	-40
XC3S250E-4TQ144I	17,58	144TQFP EP	250.000	4.896	572,00	1,20	108	221.184	0	12	4	100	-40
XC3S250E-5CP132C	17,58	132CSBGA	250.000	4.896	657,00	1,20	92	221.184	0	12	4	85	0
XC3S250E-5TQ144C	17,58	144TQFP EP	250.000	4.896	657,00	1,20	108	221.184	0	12	4	85	0
XC3S200A-4FG320I	18,12	320FBGA	200.000	3.584	667,00	1,20	248	294.912	0	16	4	100	-40
XC3S200AN-4FTG256I	18,18	256FTBGA	200.000	3.584	667,00	1,20	195	294.913	4	16	4	100	-40
XC3S400A-4FT256I	18,24	256FTBGA	400.000	7.168	667,00	1,20	195	368.640	0	20	4	100	-40
XC3S250E-4PQ208C	18,30	208PQFP	250.000	4.896	572,00	1,20	158	221.184	0	12	4	85	0
XC3S400A-4FG320C	18,66	320FBGA	400.000	7.168	667,00	1,20	251	368.640	0	20	4	85	0
XC3S200A-5FG320C	18,96	320FBGA	200.000	3.584	770,00	1,20	248	294.912	0	16	4	85	0
XC3S200AN-5FTG256C	18,96	256FTBGA	200.000	3.584	770,00	1,20	195	294.914	4	16	4	85	0
XC3S400A-5FT256C	19,02	256FTBGA	400.000	7.168	770,00	1,20	195	368.640	0	20	4	85	0
XC3S400-4TQ144C	19,10	144TQFP	400.000	7.168	630,00	1,20	97	294.912	0	16	4	85	0
XA3S250E-4VQG100Q	19,20	100VTQFP	250.000	4.896	573,00	1,20	66	221.184	0	12	4	125	-40
XA3S250E-4CPG132I	19,50	132CSBGA	250.000	4.896	572,00	1,20	92	221.184	0	12	4	100	-40
XA3S250E-4TQG144I	19,50	144TQFP EP	250.000	4.896	572,00	1,20	108	221.184	0	12	4	100	-40
XC3S250E-4FT256C	20,22	256FTBGA	250.000	4.896	572,00	1,20	172	221.184	0	12	4	85	0
XC3S200-4PQ208C	20,25	208PQFP	200.000	3.840	630,00	1,20	141	221.184	0	12	4	85	0
XC3S200-4PQ208I	20,25	208PQFP	200.000	3.840	630,00	1,20	141	221.184	0	12	4	100	-40
XC3S200-5PQ208C	20,25	208PQFP	200.000	3.840	725,00	1,20	141	221.184	0	12	4	85	0
XC3S500E-4CP132C	20,76	132CSBGA	500.000	9.312	572,00	1,20	92	368.640	0	20	4	85	0

Modelo	Precio	Encapsulado	Puertas	LUTs_FF	Fint_MHz	V	I/O	RAM_bits	Flash_bits	Mult_18x18	Ges_clk	T_MAX	T_MIN
XC3S500E-4VQ100C	20,76	100VTQFP	500.000	9.312	572,00	1,20	66	368.640	0	20	4	85	0
XC3S250E-4PQ208I	21,06	208PQFP	250.000	4.896	572,00	1,20	158	221.184	0	12	4	100	-40
XC3S250E-5PQ208C	21,06	208PQFP	250.000	4.896	657,00	1,20	158	221.184	0	12	4	85	0
XC3S400A-4FG320I	21,42	320FBGA	400.000	7.168	667,00	1,20	251	368.640	0	20	4	100	-40
XC3S400A-4FG400C	21,90	400FBGA	400.000	7.168	667,00	1,20	311	368.640	0	20	4	85	0
XC3S400-4TQ144I	21,95	144TQFP	400.000	7.168	630,00	1,20	97	294.912	0	16	4	100	-40
XC3S400-5TQ144C	21,99	144TQFP	400.000	7.168	725,00	1,20	97	294.912	0	16	4	85	0
XA3S250E-4CPG132Q	22,44	132CSBGA	250.000	4.896	572,00	1,20	92	221.184	0	12	4	125	-40
XA3S250E-4TQG144Q	22,44	144TQFP EP	250.000	4.896	572,00	1,20	108	221.184	0	12	4	125	-40
XC3S700A-4FT256C	22,50	256FTBGA	700.000	11.776	667,00	1,20	161	368.640	0	20	8	85	0
XC3S250E-4FT256I	23,28	256FTBGA	250.000	4.896	572,00	1,20	172	221.184	0	12	4	100	-40
XC3S250E-5FT256C	23,28	256FTBGA	250.000	4.896	657,00	1,20	172	221.184	0	12	4	85	0
XA3S250E-4PQG208I	23,34	208PQFP	250.000	4.896	572,00	1,20	158	221.184	0	12	4	100	-40
XC3S400A-5FG320C	23,38	320FBGA	400.000	7.168	770,00	1,20	251	368.640	0	20	4	85	0
XC3S500E-4CP132I	23,88	132CSBGA	500.000	9.312	572,00	1,20	92	368.640	0	20	4	100	-40
XC3S500E-4VQ100I	23,88	100VTQFP	500.000	9.312	572,00	1,20	66	368.640	0	20	4	100	-40
XC3S500E-5CP132C	23,88	132CSBGA	500.000	9.312	657,00	1,20	92	368.640	0	20	4	85	0
XA3S500E-4CPG132I	24,36	132CSBGA	500.000	9.312	572,00	1,20	92	368.640	0	20	4	100	-40
XC3S500E-4PQ208C	24,90	208PQFP	500.000	9.312	572,00	1,20	158	368.640	0	20	4	85	0
XC3S400AN-4FGG400C	25,14	400FBGA	400.000	7.168	667,00	1,20	311	368.640	4	20	4	85	0
XC3S400A-4FG400I	25,20	400FBGA	400.000	7.168	667,00	1,20	311	368.640	0	20	4	100	-40
XC3S400-4PQ208C	25,45	208PQFP	400.000	7.168	630,00	1,20	141	294.912	0	16	4	85	0
XC3S400-4PQ208I	25,45	208PQFP	400.000	7.168	630,00	1,20	141	294.912	0	16	4	100	-40
XC3S400-5PQ208C	25,45	208PQFP	400.000	7.168	725,00	1,20	141	294.912	0	16	4	85	0
XC3S200-4FT256C	25,55	256FTBGA	200.000	3.840	630,00	1,20	173	221.184	0	12	4	85	0
XC3S200-4FT256I	25,55	256FTBGA	200.000	3.840	630,00	1,20	173	221.184	0	12	4	100	-40
XC3S200-5FT256C	25,55	256FTBGA	200.000	3.840	725,00	1,20	173	221.184	0	12	4	85	0

Modelo	Precio	Encapsulado	Puertas	LUTs_FF	Fint_MHz	V	I/O	RAM_bits	Flash_bits	Mult_18x18	Ges_clk	T_MAX	T_MIN
XA3S250E-4FTG256I	25,80	256FTBGA	250.000	4.896	572,00	1,20	172	221.184	0	12	4	100	-40
XC3S700A-4FT256I	25,86	256FTBGA	700.000	11.776	667,00	1,20	161	368.640	0	20	8	100	-40
XC3S400A-5FG400C	26,28	400FBGA	400.000	7.168	770,00	1,20	311	368.640	0	20	4	85	0
XA3S250E-4PQG208Q	26,82	208PQFP	250.000	4.896	572,00	1,20	158	221.184	0	12	4	125	-40
XC3S400-4FT256C	27,00	256FTBGA	400.000	7.168	630,00	1,20	173	294.912	0	16	4	85	0
XC3S400-4FT256I	27,00	256FTBGA	400.000	7.168	630,00	1,20	173	294.912	0	16	4	100	-40
XC3S400-5FT256C	27,00	256FTBGA	400.000	7.168	725,00	1,20	173	294.912	0	16	4	85	0
XC3S700A-5FT256C	27,00	256FTBGA	700.000	11.776	770,00	1,20	161	368.640	0	20	8	85	0
XC3S500E-4FT256C	27,42	256FTBGA	500.000	9.312	572,00	1,20	190	368.640	0	20	4	85	0
XA3S500E-4CPG132Q	28,02	132CSBGA	500.000	9.312	572,00	1,20	92	368.640	0	20	4	125	-40
XC3S500E-4PQ208I	28,62	208PQFP	500.000	9.312	572,00	1,20	158	368.640	0	20	4	100	-40
XC3S500E-5PQ208C	28,62	256FTBGA	500.000	9.312	657,00	1,20	158	368.640	0	20	4	85	0
XC3S400AN-4FGG400I	28,92	400FPBGA	400.000	7.168	667,00	1,20	311	368.640	4	20	4	100	-40
XA3S500E-4PQG208I	29,22	208PQFP	500.000	9.312	572,00	1,20	158	368.640	0	20	4	100	-40
XA3S200A-4FTG256I	29,58	256FTBGA	200.000	3.584	572,00	1,20	195	294.912	0	16	4	100	-40
XA3S250E-4FTG256Q	29,64	256FTBGA	250.000	4.896	572,00	1,20	172	221.184	0	12	4	125	-40
XC3S400AN-5FGG400C	30,18	400FPBGA	400.000	7.168	770,00	1,20	311	368.640	4	20	4	85	0
XC3S700A-4FG400C	30,36	400FBGA	700.000	11.776	667,00	1,20	311	368.640	0	20	8	85	0
XC3S500E-4FG320C	31,56	320FBGA	500.000	9.312	572,00	1,20	232	368.640	0	20	4	85	0
XC3S500E-4FT256I	31,56	320FBGA	500.000	9.312	572,00	1,20	190	368.640	0	20	4	100	-40
XC3S500E-5FT256C	31,56	320FBGA	500.000	9.312	657,00	1,20	190	368.640	0	20	4	85	0
XA3S500E-4FTG256I	32,16	256FTBGA	500.000	9.312	572,00	1,20	172	368.640	0	20	4	100	-40
XA3S200A-4FTG256Q	33,12	256FTBGA	200.000	3.584	573,00	1,20	195	294.912	0	16	4	125	-40
XA3S500E-4PQG208Q	33,60	208PQFP	500.000	9.312	572,00	1,20	158	368.640	0	20	4	125	-40
XC3S700A-4FG484C	33,78	484FBGA	700.000	11.776	667,00	1,20	372	368.640	0	20	8	85	0
XC3S1400A-4FT256C	34,14	256FTBGA	1.400.000	22.528	667,00	1,20	161	589.824	0	32	8	85	0
XA3S400A-4FTG256I	34,70	256FTBGA	400.000	7.168	575,00	1,20	195	368.640	0	20	4	100	-40

Modelo	Precio	Encapsulado	Puertas	LUTs_FF	Fint_MHz	V	I/O	RAM_bits	Flash_bits	Mult_18x18	Ges_clk	T_MAX	T_MIN
XC3S700A-4FG400I	34,92	400FBGA	700.000	11.776	667,00	1,20	311	368.640	0	20	8	100	-40
XC3S400-4FG320C	35,70	320FBGA	400.000	7.168	630,00	1,20	221	294.912	0	16	4	85	0
XC3S500E-4FG320I	36,30	320FBGA	500.000	9.312	572,00	1,20	232	368.640	0	20	4	100	-40
XC3S500E-5FG320C	36,30	320FBGA	500.000	9.312	657,00	1,20	232	368.640	0	20	4	85	0
XC3S700A-5FG400C	36,48	400FBGA	700.000	11.776	770,00	1,20	311	368.640	0	20	8	85	0
XA3S500E-4FTG256Q	36,96	256FTBGA	500.000	9.312	572,00	1,20	172	368.640	0	20	4	125	-40
XC3S700AN-4FGG484C	37,14	484FBGA	700.000	11.776	667,00	1,20	372	368.640	8	20	8	85	0
XC3S1200E-4FT256C	38,04	256FTBGA	1.200.000	17.344	572,00	1,20	232	516.096	0	28	8	85	0
XC3S700A-4FG484I	38,82	484FBGA	700.000	11.776	667,00	1,20	372	368.640	0	20	8	100	-40
XA3S400A-4FTG256Q	38,88	256FTBGA	400.000	7.168	576,00	1,20	195	368.640	0	20	4	125	-40
XC3S1400A-4FT256I	39,30	256FTBGA	1.400.000	22.528	667,00	1,20	161	589.824	0	32	8	100	-40
XC3S700A-5FG484C	40,50	484FBGA	700.000	11.776	770,00	1,20	372	368.640	0	20	8	85	0
XC3S1400A-5FT256C	40,98	256FTBGA	1.400.000	22.528	770,00	1,20	161	589.824	0	32	8	85	0
XC3S400-4FG320I	41,05	320FBGA	400.000	7.168	630,00	1,20	221	294.912	0	16	4	100	-40
XC3S400-5FG320C	41,05	320FBGA	400.000	7.168	725,00	1,20	221	294.912	0	16	4	85	0
XC3S700AN-4FGG484I	42,72	484FBGA	700.000	11.776	667,00	1,20	372	368.640	8	20	8	100	-40
XC3S1000-4FG320C	43,00	320FBGA	1.000.000	15.360	630,00	1,20	221	442.368	0	24	4	85	0
XC3S1200E-4FG320C	43,74	256FTBGA	1.200.000	17.344	657,00	1,20	250	516.096	0	28	8	85	0
XC3S1200E-4FT256I	43,74	256FTBGA	1.200.000	17.344	572,00	1,20	232	516.096	0	28	8	100	-40
XC3S1200E-5FT256C	43,74	256FTBGA	1.200.000	17.344	572,00	1,20	232	516.096	0	28	8	85	0
XC3S1000-5FT256I	43,90	256FTBGA	1.000.000	15.360	725,00	1,20	173	442.368	0	24	4	100	-40
XC3S700AN-5FGG484C	44,58	484FBGA	700.000	11.776	770,00	1,20	372	368.640	8	20	8	85	0
XC3S1000-4FT256C	46,60	256FTBGA	1.000.000	15.360	630,00	1,20	173	442.368	0	24	4	85	0
XC3S1000-4FT256I	46,60	256FTBGA	1.000.000	15.360	630,00	1,20	173	442.368	0	24	4	100	-40
XC3S1000-5FT256C	46,60	256FTBGA	1.000.000	15.360	725,00	1,20	173	442.368	0	24	4	85	0
XC3S400-4FG456C	47,05	456FBGA	400.000	7.168	630,00	1,20	264	294.912	0	16	4	85	0
XC3S400-4FG456I	47,05	456FBGA	400.000	7.168	630,00	1,20	264	294.912	0	16	4	100	-40

Modelo	Precio	Encapsulado	Puertas	LUTs_FF	Fint_MHz	V	I/O	RAM_bits	Flash_bits	Mult_18x18	Ges_clk	T_MAX	T_MIN
XC3S400-5FG456C	47,05	456FBGA	400.000	7.168	725,00	1,20	264	294.912	0	16	4	85	0
XC3S1000-4FG320I	47,18	320FBGA	1.000.000	15.360	630,00	1,20	221	442.368	0	24	4	100	-40
XA3S400A-4FGG400I	48,06	400FBGA	400.000	7.168	574,00	1,20	311	368.640	0	20	4	100	-40
XA3S1200E-4FTG256I	48,48	256FTBGA	1.200.000	17.344	572,00	1,20	190	368.640	0	28	8	100	-40

Tabla 34. Listado de FPGAs de las serie Spartan3 de Xilinx con precio inferior a 50$

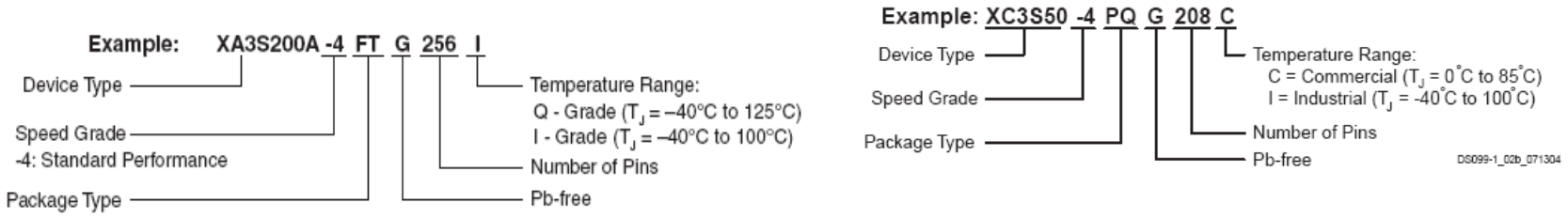


Figura 34. Nomenclatura empleada por Xilinx para identificar a las Spartan3

Suman un total de 246 modelos, 81 de Altera y 165 de Xilinx. Al tratarse de una muestra filtrada por precio, se observa la heterogeneidad de sus características. Así aparecen FPGAs con distintos grados de velocidad, encapsulados, temperatura, recursos lógicos, etc.

El PdC no tratará de comparar el comportamiento de ciertos circuitos usando como objetivos FPGAs similares, sino que tratará de establecer un orden entre ellas atendiendo al precio como principal índice de referencia.

3.6. Herramientas de diseño con FPGAs

Para ejecutar y analizar el PdC se hacer necesario contar con herramientas específicas para diseño con FPGAs. Tras concluir cada una de las etapas de diseño, se dispondrá de una serie de ficheros de texto con información sobre los resultados de las fases de síntesis, implementación así como de los análisis de tiempo y potencia. Actualmente este tipo de software es ofrecido al mercado como paquetes integrados tanto por los propios fabricantes de FPGAs como por terceras empresas. Para la ejecución y revisión del PdC de este trabajo deben incluir los siguientes módulos:

- Herramienta de Síntesis Lógica
- Herramienta de Implementación
- Herramienta de Análisis Estático de Tiempo (STA)
- Herramienta de Análisis de Potencia

Para la **síntesis lógica** es fácil encontrar, junto a las ofrecidas por las marcas de FPGAs, una amplia oferta de terceras compañías:

- LeonardoSpectrum y Precision (RTL / Physical) de Mentor Graphics
- Synplify de Synopsis
- Quartus II (entorno integrado) de Altera
- XST (incluido en el paquete ISE) by Xilinx
- IspLever de Lattice Semiconductor
- Libero de Actel

Las **herramientas de implementación** se ocupan de las fases de traducción, mapeado y emplazamiento/ruteado. No hay más opción disponible que las ofrecidas por los fabricantes: Xilinx, Altera, Actel, Lattice…

Los **analizadores estáticos de tiempo** estudian el comportamiento temporal de los diseños y forman parte de los paquetes integrados ofrecidos por las marcas ya conocidas. Este tipo de herramientas ha desplazado últimamente a los analizadores dinámicos, ya que cada vez es más difícil generar vectores de test dada la extensión y complejidad de los proyectos actuales. La verificación basada en análisis estáticos juega un papel importante facilitando una rápida y razonable medida de tiempo del camino crítico de un circuito. En algunos casos, no necesita que el diseñador especifique restricciones a las que ajustarse, imponiendo la propia herramienta algunas. Todo esto permite reducir el tiempo dedicado a la verificación del diseño y prescindir casi totalmente de la intervención del usuario.

Dada la creciente complejidad de los diseños y la reducción de los procesos geométricos, es habitual trabajar con grandes frecuencias. Esta situación, da lugar a que el **análisis de potencia** no se centre solo en el consumo de cara a la alimentación del circuito, sino también en el calor generado para determinar una correcta refrigeración del dispositivo programable. Reguladores de tensión, disipadores, pastas térmicas, etc., serán elementos que incrementarán el coste final del sistema que aloje la FPGA seleccionada.

Para realizar el análisis de consumo y las consideraciones térmicas, existen dos tipos de herramientas: de estimación y de evaluación. El primer tipo, permite estimar el consumo de un determinado diseño sin necesidad de disponer de una lista de conexiones, o *netlist*, del circuito. El segundo, evalúa la potencia del diseño ya implementado que será tanto más fiable cuanto más completa sea la información que el usuario

proporcione. Aún así, los fabricantes recomiendan usar estos resultados solo como guías y medir el consumo en el producto final.

Estas herramientas basan sus cálculos en dos modelos de potencia, el estático y el dinámico. La potencia estática está originada por la corriente de fuga existente en los semiconductores incluso en estado de reposo, depende de la temperatura de la unión y del tamaño del dado entre otros parámetros. La potencia dinámica es consumida cuando un dispositivo está en funcionamiento debido a la conmutación de las señales por la carga y descarga de las capacidades de los elementos conectados a las entradas y salidas. Depende por tanto del valor de la capacidad citada, de la frecuencia de conmutación y de la tensión de trabajo, Ecuación 6. La suma de ambas será la potencia total consumida por el circuito.

$$\boldsymbol{P_{dinámica} = C_L * V_{cc}^2 * F_{output}} \quad \textbf{(Ecuación 6)}$$

La evaluación de potencia se completa con un estudio térmico para el dispositivo y encapsulado seleccionados. Por defecto, a partir de los datos de catalogo, devuelven información sobre la resistencia térmica (θ) existente en el interfaz unión-aire (θ_{JA}), Ecuación 7, y calculan la temperatura de la unión para un determinado valor del flujo de aire y de la temperatura ambiente. Es posible incluir datos sobre los disipadores previstos y modificar las condiciones de trabajo.

$$\boldsymbol{\theta_{unión-aire} = \theta_{unión-encapsulado} + \theta_{encapsulado-aire}} \quad \textbf{(Ecuación 7)}$$

Todas las FPGAs seleccionadas para este trabajo se encuentran disponibles como tecnologías objetivo en las versiones gratuitas

de los entornos integrados ofrecidos por los fabricantes de FPGA. Ninguna presenta limitaciones de tamaño o velocidad que influyan en los resultados de las implementaciones de los circuitos patrones. Tampoco restringen su uso industrial e incluyen todos los bloques necesarios: síntesis lógica, implementación, análisis estático de tiempo y análisis de potencia. Se pretende demostrar su validez para obtener buenos resultados con diseños sencillos sobre las FPGAs más económicas.

En la Tabla 35 se recogen las herramientas ofrecidas por los cuatro fabricantes de LP inicialmente considerados. Se destaca en negrita los que finalmente han sido elegidos para la ejecución del PdC que se propone. Se ha optado solo por los dos fabricantes que ofrecen una versión gratuita de sus entornos de trabajo dado que esta cualidad redunda en el coste del binomio FPGA-herramienta.

Fabricante	**Altera Corp.**	**Xilinx Inc.**	Actel Corp.	LatticeSemi
Nombre del entorno de desarrollo	**Quartus II**	**ISE**	Libero IDE	ispLever
Nombre herramienta gratuita	**Quartus II Web Edition**	**WebPack ISE**	Libero IDE (Limitada 1 año)	ispLever Starter (Limitada a 6 meses)
Versión actual (gratuita)	**9.0**	**10**	8.5	7.2

Tabla 35. Entornos de desarrollo de distribución gratuita

Las versiones gratuitas, no cuentan con algunas de las herramientas más complejas ni son válidas para tecnologías más avanzadas. Por ejemplo, Altera en Web Edition no incluye la familia Stratix ni herramientas como la compilación incremental. Webpack de Xilinx permite trabajar con casi todas las Spartan3 y un gran número de modelos de Virtex 4 y 5 aunque no incluye los módulos específicos para diseño embebido o DSP. Actel dispone de

licencia de evaluación para Libero IDE v.8.5, para un periodo de cuarenta y cinco días con todas sus características al completo, y otra de un año que solo incluye soporte para dispositivos de menos de 1.500 puertas. Lattice Semiconductor ofrece una versión inicial gratuita, ispLever Starter, para un período de de seis meses.

3.6.1. Configuración de las herramientas de diseño

Para conseguir tiempos de mercado cortos, es interesante que una vez finalizada la fase de diseño, los procesos de síntesis, emplazamiento e interconexionado se ejecuten de forma inmediata, esto es, no tener que ejecutar múltiples iteraciones para conseguir un buen rendimiento. Este último camino es el que recomienda Xilinx para su software según sus tutoriales.

Las versiones más modernas de las herramientas ofrecen al usuario configuraciones predeterminadas agrupadas bajo el nombre de "estrategia". Proponen unas cuantas estrategias con diferentes objetivos y establecen por defecto una, denominada "Balanced", que configura a la herramienta para conseguir un equilibrio entre velocidad, recursos lógicos ocupados o tiempo de ejecución.

Herramienta	Webpack ISE 10	QuartusII WebEdition 9.0
Estrategia por defecto	**Balanced**: ofrece una combinación equilibrada de rendimiento y tiempo de ejecución.	**Balanced:** combina una implementación de alto rendimiento y una utilización de recursos baja.
Otras estrategias	**Minimun runtime:** Se centra en conseguir tiempos de ejecución reducidos.	

Herramienta	Webpack ISE 10	QuartusII WebEdition 9.0
	Timing performance: Dado un valor de tiempo objetivo, la herramienta trata de ajustarse lo máximo posible a ese valor.	**Speed:** El compilador consigue la implementación de mayor frecuencia
	Power optimization: Reduce el consume de potencia del diseño	
		Área: el compilador reduce al máximo al diseño para minimizar la utilización de recursos.

Tabla 36. Estrategias propuestas por cada fabricante en sus herramientas

Indiscutiblemente, un profundo conocimiento de las herramientas y una adecuada configuración específica por cada diseño, da lugar a los mejores resultados. Sin embargo, las estrategias recomendadas por cada fabricante se entienden como un buen punto de partida para llegar a una futura optimización. Es de suponer, que la estrategia balanceada ofrecida en todos los entornos integrados, consigue una implementación adecuada, con un buen rendimiento y en un tiempo prudencial.

Es interesante considerar tanto los resultados devueltos por esta estrategia de equilibrio como por otras de las existentes, y comprobar si el parámetro frecuencia mejora considerablemente en aquellas cuyo objetivo es mejorar los tiempos de trabajo del circuito implementado.

Este trabajo hará uso de la **estrategia *Balanced*** y otra común en todas las herramientas, en concreto, aquella que trata de conseguir la mejor frecuencia interna del circuito (*Timing performance* ó *Speed*, según fabricante).

Para más detalle sobre las opciones de configuración preestablecidas por cada fabricante para las tareas de síntesis, mapeado, emplazamiento e interconexiones, ver el anexo A.

3.7. Consideraciones

Las FPGAs han ido evolucionando en flexibilidad y reducción de costes a medida que se iban sumando nuevos mercados donde pueden ver proyectados en ellas sus productos como sistemas en un chip. El capítulo muestra a los sectores de la **automoción**, **la electrónica de consumo y de las comunicaciones** como los grandes beneficiados de las virtudes técnicas y económicas de las modernas FPGAs de bajo coste y a ellos se dirigirán principalmente los resultados de la propuesta de PdC.

El diseñador, que agradece la variedad pero que se asusta ante la ardua tarea de selección que le espera, y quien al margen de las restricciones impuestas por el propio diseño: **número de entradas/salidas, temperatura de trabajo**, etc., debe valorar otros parámetros con igual importancia en infinidad de modelos y sin perder de vista el objetivo de minimizar costes. Y es en este punto donde se detecta una carencia de alguna herramienta que oriente en esta tarea, al menos, para realizar una primera criba que devuelva un conjunto más reducido de modelos conforme a unas necesidades.

La revisión de las múltiples arquitecturas consideradas sirve para identificar los elementos comunes de las FPGAs que centrarán este estudio e identificarlos como parámetros de interés de cara a su clasificación, m**ultiplicadores y bloques de memoria** se revelan como nuevos indicadores del área de ocupación ed un diseño.

Por otro lado, el capitulo pretende ayudar a establecer una valoración cualitativa ante la pléyade de FPGAs comercializadas, centrándose en características difíciles de medir, concluyéndose:

- Cuales son los fabricantes que ofrecen **herramientas gratuitas: Xilinx y Altera**
- Cuáles disponen de **núcleos virtuales** de microprocesadores: **Xilinx, Altera, Actel y Lattice**
- Cuáles permiten **configuración tipo flash: Actel y Xilinx**.
- Qué fabricante ofrecen mayor **garantía de continuidad en el suministro** del producto elegido. La información financiera de las empresas presenta una perspectiva económica y comercial que permiten concluir que la presencia dominante de **Xilinx y Altera** es un aval de continuidad.

Este capítulo ha permitido satisfacer los objetivos a) y b) expuestos al principio del documento:

"a) Identificar los principales sectores comerciales que hacen un mayor uso de electrónica programable del tipo FPGA."

"b) Concretar qué tipos de dispositivos programables FPGA presentaron una mayor demanda en los últimos años para incluirlas o no como tecnologías objetivo en los test a realizar. "

Pone en evidencia la utilidad que tendría disponer de un asistente que permitiese al diseñador pilotar su búsqueda optimizándola de acuerdo a unos objetivos y conduce el trabajo hacia el siguiente capitulo.

4. MÉTODO DE CREACIÓN Y EJECUCIÓN DEL PDC

En la búsqueda de un método operativo para obtener un PdC, huiremos de un método exhaustivo a favor de uno efectivo, de un lado un método infalible para determinar qué modelo de FPGA se comporta mejor alojando un determinado diseño, sería ejecutar la implementación más óptima del mismo sobre todas y cada unas de las tecnologías existentes. Aunque se trata de un método certero aduce de ser lento y arduo ya que exige mucho tiempo de trabajo, un gran conocimiento de las tecnologías existentes y una amplia experiencia con las herramientas de diseño.

Alternativamente para estudiar y puntuar cada FPGA se propone utilizar módulos IPs frecuentemente utilizados en los sectores de mercado ya citados. Dada la imposibilidad de emplear módulos propietarios, una buena opción es acudir al mayor repositorio de núcleos IP públicos: OpenCores, siguiendo la pauta marcada por Altera en su última propuesta de benchmark (23).

Como se trata de evaluar FPGAs de bajo coste, los circuitos elegidos para este benchmark no serán solo y exclusivamente los más populares como utilizó Altera, esto es, los que han tenido mayor número de accesos según OpenCores. La propuesta pasa por seleccionar aquellos núcleos más representativos de los sectores de la automoción, comunicaciones y electrónica de consumo y que cumplan con las características que más adelante se detallan.

4.1. El consorcio Opencores

Esta organización fue fundada en 1999 por Damjan Lampret como una comunidad de código abierto para el desarrollo y distribución de IPs descritos mediante los lenguajes VHDL y Verilog. Estudiantes, investigadores e ingenieros entre otros comparten su conocimiento a través de este repositorio de gran prestigio internacional que nació al amparo de la filosofía "software abierto" y hoy día se presentan como promotores del "hardware abierto".

Su página de internet organiza todos los módulos funcionalmente, aporta información sobre el estado de desarrollo de cada uno, incluye documentación de soporte y permite la descarga directa de todos los componentes del proyecto. A la fecha Opencores cuenta con unos 500 proyectos en su servidor central de Suecia, pero la lista no para de crecer.

4.1.1. Licencia Opencores

En 1985, Richard Stallman funda la *Free software foundation* (FSF) e inicia el proyecto GNU para desarrollar software libre y colaborativo. La FSF mantiene y defiende la Licencia Pública General GNU (GNU GPL) y la Licencia Pública General Reducida GNU (GNU LGPL o LesserGPL). Surgieron otras iniciativas en esta línea como la BSD promulgada por la Universidad de California Berkeley de la cual existe una actualización denominada BSD modificada. Y la lista puede continuar con un gran número de licencias que siguen estas ideas.

Dado que Opencores surge con la misma filosofía que los anteriores movimientos, hace uso de estas licencias para "proteger" sus trabajos. Cada autor que aporta módulos IP

utiliza la que considera más conveniente y la especifica en las características de su proyecto. Recientemente, el consorcio ha empezado a recomendar a los desarrolladores que se acojan solo a dos tipos de licencia: Lesser GPL (LGPL) o bien BSD modificada.

La licencia BSD no ofrece garantía, el código debe ser distribuido en forma binaria o fuente y los autores deben ser reconocidos. GNU Lesser(LGPL) surge como un compromiso entre la restrictiva GPL original y la permisiva BSD. Como ventaja, aporta la posibilidad de que un código GPL sea utilizado como librería por otro programa no GPL.

4.2. Requisitos de los IPs-Opencores para el PdC

La selección de IPs de Opencores atiende a una serie de condiciones que a continuación se citan. Este filtrado consigue reducir el conjunto de proyectos disponibles, manteniendo solo aquellos que permitan una comparación equitativa cuando se ejecute el PdC sobre el grupo de FPGAs a experimentar.

1ª Condición: Que el diseño sea totalmente síncrono

Todos los componentes del circuito obedecerán al flanco activo de una única señal de reloj cambiando de estado en el mismo instante. Las razones por las que se sigue esta condición son:

- Las FPGAs cuentan con una arquitectura orientada a este tipo de funcionamiento, por tanto los diseños síncronos que sobre ellas se implementen devolverán mejores resultados.
- Los diseños síncronos son más fáciles de verificar y son más fiables siempre que la frecuencia y el sesgo del reloj empleado no superen los límites admisibles en cada FPGA.

- La mayoría de las FPGAs incluyen biestables activados por flancos
- Aunque existen diseños síncronos con más de una señal de reloj, el tratamiento de los mismos requiere una sincronización de las transferencias de datos de los diferentes dominios y dificulta el traslado de un diseño de una arquitectura a otra. Esta circunstancia daría lugar a que cada herramienta les diera un trato diferente. Por tanto, los resultados no serían comparables salvo configurando cada herramienta con parámetros diferentes a los establecidos por defecto (alta intervención del usuario), lo que se alejaría de unos de los objetivos perseguidos.

2ª Condición: El reloj del circuito debe distribuirse por todo el dispositivo empleando las líneas de interconexión global existentes en todas las FPGAs.

Mediante esta condición se garantiza un mínimo sesgo al tener estas líneas un retardo de propagación mínimo.

3ª Condición. Que los circuitos elegidos no hagan uso estructuras complejas de reloj.

Ya que los analizadores de tiempo de cada herramienta les dan un tratamiento diferente (22). Los IP incluidos en este PdC no tendrán estructuras de reloj de los siguientes tipos:

- Reloj registrado: entrada de reloj manejada por la salida de otro elemento secuencial.

- Reloj con lógica: relojes que atraviesan algún elemento lógico. Puede añadir sesgo entre los registros afectados por ese reloj.
- Gestores de reloj como DCM, DLL o PLL.
- Bucles combinacionales: la salida de una estructura lógica se emplea como entrada a la misma estructura. El retardo lógico deberá ser considerado por partida doble.

4ª Condición. Utilizar diseños reducidos.

Dado que los modelos que se usarán como objetivo para la implementación serán FPGAs de menos de 50 $ por unidad, su capacidad lógica y el número de entradas/salidas disponibles por el usuario serán muy reducidos limitando los diseños de gran tamaño.

4.3. Procedimiento de obtención y propuesta de PdC

La lista de proyectos disponibles en Opencores se ordena funcionalmente ofreciendo la posibilidad de establecer algunos criterios de filtrado, Figura 35.

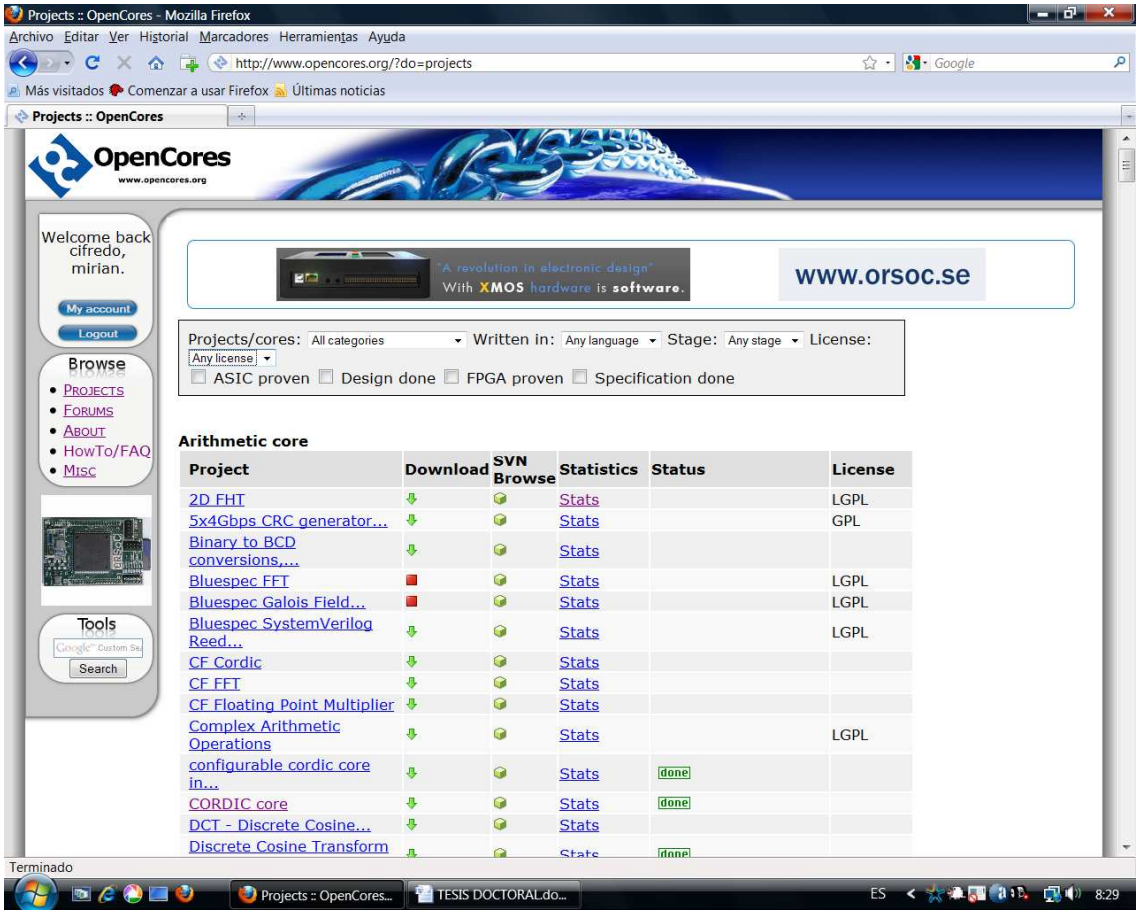


Figura 35. Imagen capturada del sitio de internet www.opencores.org

Los pasos seguidos para determinar una lista con un número de IPs adecuados y conformes a los criterios anteriormente expuestos son:

1. Para el campo *Projects/Core* seleccionar el criterio *All categories*, marcar la casilla correspondiente a *Design Dones* y asignar a la lista desplegable *Stage* la opción *Stable,* Figura 36. El resultado es una lista con todos los proyectos que han sido valorados como *DONE*. Es preferible prescindir de aquellos que no tienen el visto bueno o que no son versión definitiva (Alpha o Beta), ya que al estar aún en fase de prueba no garantizan un correcto funcionamiento.

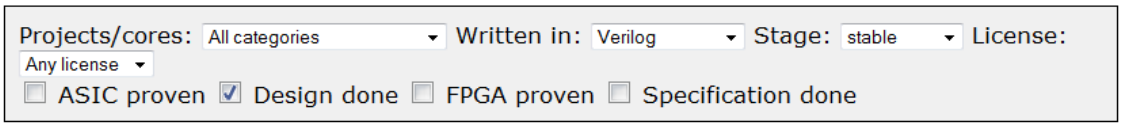


Figura 36. Aspecto de la tabla de búsqueda de Opencores

2. Inicialmente se establecen dos listas: una con descripciones en VHDL y otra con Verilog. Nuevamente se hace uso de los criterios de búsqueda disponibles en el interfaz ofrecido por Opencores extrayendo una primera lista de circuitos con un total de 46 en VHDL y 31 en Verilog.

Arithmetic Cores

Project	Download	SVN	Browse	Statistics	Status	License
CORDIC core				Stats	done	
Discrete Cosine Transform core				Stats	done	
Floating Point Adder and...				Stats	done	
Single Clock Unsigned...				Stats	done	
Sinusoidal Computator				Stats	done	LGPL

Prototype board

Project	Download	SVN	Browse	Statistics	Status	License
MAXII-Evalboard				Stats	done	
USB FPGA MODULE 1.2				Stats	done	GPL

Communication controller

Project	Download	SVN Browse	Statistics	Status	License
8b10b Encoder/Decoder			Stats	done	GPL
a VHDL 16550 UART core			Stats	done	
A VHDL CAN Protocol Controller			Stats	done	
I2S Interface			Stats	done wbc	GPL
SD/MMC Bootloader			Stats	done	GPL
Simple AES3 / SPDIF receiver			Stats	done	LGPL
SPDIF Interface			Stats	done wbc	LGPL
spi master receiver for ADC...			Stats	done	LGPL
UART16750			Stats	done	LGPL

Crypto Cores

Project	Download	SVN Browse	Statistics	Status	License
AES core modules			Stats	done	
AES128			Stats	done	LGPL
Avalon AES ECB-Core (128,...			Stats	done	BSD
twofish 128/192/256			Stats	done	GPL

DSP Cores

Project	Download	SVN Browse	Statistics	Status	License
DDS Synthesizer			Stats	done	GPL
Hilbert Transformer			Stats	done	GPL

ECC Cores

Project	Download	SVN Browse	Statistics	Status	License
Ultimate CRC			Stats	done	GPL

Library

Project	Download	SVN Browse	Statistics	Status	License
extension_pack			Stats	done	
gh vhdl library			Stats	done	
Random Number Generator Library			Stats	done	GPL

Memory Core

Project	Download	SVN Browse	Statistics	Status	License
srl_fifo			Stats	done	LGPL

Processor

Project	Download	SVN Browse	Statistics	Status	License
HC11 Compatible - Gator...			Stats	done	
LEM1_9			Stats	done	LGPL
Plasma - most MIPS I(TM) opcodes			Stats	done	Others
T400 µController			Stats	done	GPL
T48 µController			Stats	done	GPL
ZPU - the worlds smallest 32...			Stats	done wbc	

Other

Project	Download	SVN Browse	Statistics	Status	License
a VHDL 8254 Timer			Stats	done	
Hardware looping unit			Stats	done	GPL
IMA ADPCM Sound Encoder			Stats	done wbc	GPL

SoC

Project	Download	SVN Browse	Statistics	Status	License
ahb system generator			Stats	done	LGPL
GECKO3 SoC co-design environment			Stats	done wbc	Others
wishbone out port from b3 spec			Stats	done wbc	LGPL
Wishbone System6800/01			Stats	done wbc	

System Controller

Project	Download	SVN Browse	Statistics	Status	License
PCI Target			Stats	done wbc	LGPL

Video Controller

Project	Download	SVN Browse	Statistics	Status	License
JPEG Hardware Compressor			Stats	done	
PAL/NTSC encoder			Stats	done	LGPL
Sharp LQ057Q3DC02 LCD Controller					

Figura 37. Imagen del resultado obtenido tras aplicar los filtros (VHDL)

Prototype board

Project	Download	SVN Browse	Statistics	Status	License
kiss-board			Stats	done wbc	

Communication controller

Project	Download	SVN Browse	Statistics	Status	License
10_100_1000 Mbps tri-mode...			Stats	done	LGPL
I2C controller core			Stats	done wbc	
I2C Slave			Stats	done	GPL

sd card controller			Stats	done wbc	LGPL
SD/MMC Controller			Stats	done wbc	GPL
SPI core			Stats	done wbc	
USB 1.1 Host and Function IP...			Stats	done wbc	LGPL
Wishbone LPC Host and...			Stats	done wbc	LGPL

ECC Core

Project	Download	SVN Browse	Statistics	Status	License
Reed Solomon Decoder (204,188)			Stats	new done	GPL
Reed Solomon Encoder			Stats	done	
Reed-Solomon Decoder (31, 19, 6)			Stats	done	

Memory core

Project	Download	SVN Browse	Statistics	Status	License
Asynchronous...			Stats	done wbc	GPL
High Performance Dynamic...			Stats	done	Others
Wishbone FLASH Interface for...			Stats	done wbc	LGPL

Processors

Project	Download	SVN Browse	Statistics	Status	License
8080 Compatible CPU			Stats	done	
ClaiRISC - runs 12bit opcode...			Stats	done	
cpu6502_tc - R6502			Stats	done	GPL

Processor...					
openMSP430			Stats	new done	LGPL
OpenRISC 1000			Stats	done wbc	LGPL
S1 Core			Stats	done wbc	GPL
Wishbone High Performance Z80			Stats	done wbc	

Other

Project	Download	SVN Browse	Statistics	Status	License
8051 Slave to Wishbone Master...			Stats	done wbc	
Arbitrary Frequency Generator			Stats	done	LGPL
boundaries			Stats	done	
DPLL with a Random Walk Filter			Stats	done	LGPL
DragonBall/68K Wishbone...			Stats	done wbc	
I2S Audio Pattern Generator			Stats	done	

System Controller

Project	Download	SVN Browse	Statistics	Status	License
pci_mini			Stats	done wbc	LGPL
WB LCD Character Display...			Stats	done wbc	LGPL

Video Controller

Project	Download	SVN Browse	Statistics	Status	License
VGA/LCD Controller			Stats	done wbc	

Figura 38. Imagen de los resultados obtenidos tras aplicar los filtros (Verilog)

3. Los designados como WBC, esto es, que incluyen algún bloque descriptivo que les permite compatibilidad con el bus Wishbone (28), se descartan. De este modo se evita tanto agotar recursos lógicos implementando este circuito como la aparición de un segundo dominio de reloj.

4. Se prescinde de aquellos que no pueden ser descargados y que están marcados por un cuadrado rojo.

Estas dos etapas, 3 y 4, se realizan sin ayuda del interfaz de Opencores y reducen el listado anterior a 36 núcleos VHDL y 12 Verilog.

5. Se ejecuta la fase de síntesis con las herramientas de cada fabricante sobre las tecnologías de mayor capacidad para comprobar:

- Si hay circuitos que presenten algún tipo de fallo de sintaxis.
- Si hay que inicializar el valor de algún elemento parametrizado.
- Si necesitan de alguna librería de la que no disponen.
- Que el diseño es suficientemente pequeño para que pueda implementarse sobre alguna de las FPGAs seleccionadas.
- Que no hace uso de recursos específicos de una tecnología concreta
- Que el diseño es síncrono y solo emplea un dominio de reloj
- Que no hace uso de estructuras complejas de reloj

En las Tabla 37 y Tabla 38 se detallan las incidencias acaecidas durante la fase de síntesis de cada uno de los 48 núcleos con las dos herramientas empleadas.

CIRCUITOS	Problemas en ISE	Problemas en QUARTUS II	CUMPLE REQUISITOS
CORDIC core			SI
Discrete Cosine Transform core	Fallo de código		NO
Floating Point Adder and...			SI
Single Clock Unsigned...	Falta librería		NO
Sinusoidal Computator			SI
MAXII-Evalboard	Es el diseño de una placa con FPGA Altera		No tiene interés
USB FPGA MODULE 1.2	Es el diseño de una placa con FPGA Xilinx		No tiene interés
8b10b Encoder/Decoder	No es síncrono		NO
a VHDL 16550 UART core	Incluye circuitería adicional WishBone		NO
A VHDL CAN Protocol Controller	Incluye circuitería adicional WishBone		NO
SD/MMC Bootloader			SI
Simple AES3 / SPDIF receiver	Incluye código para el uP sw de Xilinx MicroBlaze		NO
spi master receiver for ADC...			SI
UART16750			SI
AES core modules			SI
AES128			SI
Avalon AES ECB-Core (128,...	Falta librería		NO
twofish 128/192/256	Ocupación de área grande		NO
DDS Synthesizer	Se utiliza el módulo de una frecuencia y fase		SI

CIRCUITOS	Problemas en ISE	Problemas en QUARTUS II	CUMPLE REQUISITOS
Hilbert Transformer	Válido el bloque 1ª		SI
Ultimate CRC	Hay que dar valores iniciales		Si
gh vhdl library	Librerías de pequeños circuitos básicos		No tiene interés
extension_pack	Librerías de pequeños circuitos básicos		No tiene interés
Random Number Generator Library	Librerías de pequeños circuitos básicos		No tiene interés
srl_fifo	Específico para Xilinx		No tiene interés
HC11 Compatible - Gator...	Hay que interconectar los módulos existentes. Al hacerlo se obtienen algunos fallos		NO
LEM1_9	Ninguna de las entradas son registradas		NO
Plasma - most MIPS (TM) opcodes	No es síncrono: entradas y salidas sin registrar.		NO
T400 µController			SI
T48 µController	Utiliza dos relojes		NO
a VHDL 8254 Timer	No es síncrono e incluye circuitería para wishbone		NO
Hardware looping unit		Funciona eliminando tops5 y add	No tiene interés

CIRCUITOS	Problemas en ISE	Problemas en QUARTUS II	CUMPLE REQUISITOS
ahb system generator	Requiere disponer de lenguajes de scripts para generar ciertos módulos		NO
JPEG Hardware Compressor	Específico para Xilinx		NO
PAL/NTSC encoder	Utiliza dos relojes		NO
Sharp LQ057Q3DC02 LCD Controller		Error de código	NO

Tabla 37. Incidencias de los 32 IPs VHDL durante la síntesis

CIRCUITOS	**Problemas en ISE**	**Problemas en QUARTUS II**	**CUMPLE REQUISITOS**
10_100_1000 Mbps tri-mode...	Errores de código		NO
I2C Slave			SI
Reed Solomon Decoder (204,188)			SI
Reed Solomon Encoder			SI
Reed-Solomon Decoder (31, 19, 6)	Se utiliza el modelo anterior		No tiene interés
High Performance Dynamic...	Incluye circuitería para Wishbone		NO
8080 Compatible CPU	No hay fichero TOP		NO
ClaiRISC - runs 12bit opcode...	Falta algún fichero		NO
cpu6502_tc - R6502 Processor...		Fallo de código	NO
openMSP430	Falta algún fichero		NO
Arbitrary Frequency Generator	Es un circuito PLL		No tiene interés
Boundaries	Pequeños circuitos		No tiene interés

Tabla 38. Incidencias de los 16 IPs Verilog durante la síntesis

Tras seguir estos pasos se obtiene el total de IPs que conformarán el PdC, Tabla 39. Los diseños poco relevantes en los sectores comerciales automoción, consumo y comunicaciones, son etiquetados como “No tiene interés” y se descarta su inclusión en este PdC.

Lenguaje	Nombre Opencore del Circuito	Función	Autor
Verilog	Reed Solomon Encoder	Permite corregir errores de transmisión	
	Reed Solomon Decoder	Permite corregir errores de transmisión	Varkon Semiconductors (GPL)
	I2C slave	Interfaz I2C	Fielding, Steve (GPL)
VHDL	Cordic Core (R2P)	Convierte FFT a módulo y fase	Herveille, Richard
	Sinusoidal Computer	Genera formas de onda seno	Bureriu, Cristian (LGPL)
	SD/MMC Bootloader	Gestiona la configuración y arranque de una FPGA SRAM	Laeuger, Arnim (GPL)
	Floating Point Unit	Unidad de cálculo en coma flotante	Marcus, Guillermo
	UART16750	UART 16550/16750	Witt, Sebastian (LGPL)
	AES encoder	Codificador AES 128/192/256 bit	Gbur, Jerzy
	DDS Synthesizer	NCO(number controlled oscillator). Genera onda senoidal de frecuencia y fase concretas	Kumm, Martin (GPL)
	SPI Master Receiver for ADC	Receptor SPI para un ADC similar al AD747X	Alex Rayne (LGPL)
	AES decoder	Decodificador AES	Gbur, Jerzy
	AES128	AES , clave 128 bits , modo de operación ECB.	(LGPL)
	Hilbert Transformer	Modulación/demodulación, detección de	Kumm, Martin (GPL)

Lenguaje	Nombre Opencore del Circuito	Función	Autor
		amplitud o fase.	
	Ultimate CRC	Comprobador/generador CRC	Drange, Geir (GPL)
	T400 microcontroller	Controlador de 4-bit de la familia COP400	Laeuger, Arnim (GPL)

Tabla 39. Listado final con los 16 IPs que formarán el PdC

6. El último paso consiste en agrupar los IPs finales por sector de mercado, Tabla 40, según sean más o menos utilizados por éstos. Como referencia sirven los documentos de las aplicaciones comerciales de sus FPGAs expuestas por los fabricantes en internet.

Automocion	Comunicaciones	Consumo
Sinuosidal computer (gen sen)	Sinusoidal Computer	AES Core Modules_enc
SD/MMC bootloader	DDS_Synthesizer	AES128
SPI Master Receiver for ADC	FPU	AESCoreModules_dec
AES enc	I2CSlave	Cordic Core
AES128	Hilbert transform	Sinusoidal computer
T400 uC	CRC ultimate	I2Cslave
AES dec	Reed Solomon decoder	SD/MMC Bootloader
	UART16750	Reed Solomon enc
	Reed Solomon encoder	Reed solomon dec
	SD/MMC Bootloader	CRC ultimate

Tabla 40. Composición final del PdC agrupados por sector de interés

4.4. Ejecución del PdC

Los benchmarks de FPGAs existentes hasta el momento, PREP o los propuestos por Altera optan por instanciar cada uno de los circuitos integrantes del PdC tantas veces como sea posible. Más tarde, el número de instancias que se ha conseguido incluir en cada tecnología les servirá para establecer una relación entre los recursos utilizados en cada modelo. Esta técnica permite establecer una comparación de los recursos lógicos utilizados independiente de la arquitectura subyacente de cada tecnología.

Otra de las razones de esta práctica, es la de agotar los recursos de interconexión de la FPGA. Esta circunstancia da lugar al peor caso en que puede estudiarse la frecuencia máxima y por tanto obtener un resultado menos distorsionado que con el dispositivo sin rellenar.

Para mantener independiente cada instancia incluyen flip-flops entre ellas y evalúan el comportamiento de cada una para más tarde promediar la frecuencia máxima conseguida entre todos los bloques.

Sin embargo, la inclusión de múltiples instancias del mismo circuito en un mismo proyecto y tecnología no refleja la realidad. No es habitual que un diseño se componga de varios bloques exactamente iguales, solo implementar un diseño suficientemente grande y formado por varios bloques constructivos diferentes establecería unas condiciones reales con la que medir la frecuencia máxima de un circuito.

Por eso esta tesis propone instanciar una sola vez cada circuito del PdC siendo conscientes de que la frecuencia máxima que se obtenga disminuirá si se añaden nuevos núcleos al diseño. No se trata de medir la frecuencia máxima que permite la FPGA, sino la del circuito patrón como único elemento sobre cada tecnología.

Siguiendo este método se evita agregar elementos lógicos para el aislamiento del núcleo central y los datos sobre ocupación de los informes harán referencia solo al circuito en sí.

Además, hay que tener en cuenta que algunas de las herramientas de diseño incluyen en su configuración por defecto la opción de eliminar lógica redundante, y ya que el PdC se ejecutará sin modificar ningún parámetro, replicar el mismo circuito no conduciría a una implementación múltiple del mismo.

Debe quedar claro que los resultados conseguidos con el método propuesto son solo una estimación inicial del comportamiento de cada dispositivo ensayado y que dan una visión limitada del comportamiento de las FPGAs. Éstas ofrecen características adicionales que condicionarán la elección de un modelo u otro.

4.4.1. Preparación del PdC

En este momento del trabajo se dispone de un PdC organizado en tres grupos atendiendo a los sectores comerciales previamente elegidos:

Grupo por sector comercial	Número de circuitos
Automoción	7
Consumo	10
Comunicaciones	10

Esta clasificación permite que pueda ejecutarse uno de los tres PdC si se está interesado en una clasificación sectorial. Para el caso de una evaluación global igualmente se ejecutarán los

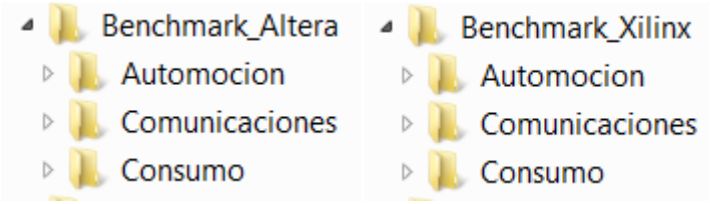
tres PdC, pero pueden eliminarse aquellos circuitos que sean comunes entre sectores.

Se recomienda seguir la metodología siguiente para adecuar los proyectos Opencores seleccionados a los comandos de cada herramienta. Siguiendo estos pasos la ejecución no ocasionará errores.

Paso 1

- Preparar dos carpetas en el disco c del ordenador personal, “Benchmark_Altera” y “Benchmark_Xilinx”, para recibir el conjunto de circuitos integrantes del PdC determinado.

- Dentro de éstas, crear otras tres con el mismo nombre de los sectores objetivos del trabajo.



Paso 2

Dentro de la carpeta de cada sector, crear otras nuevas, una por cada circuito a probar. El nombre de estas carpetas viene determinado por el de la entidad de mayor jerarquía en el diseño. En la Tabla 41 se muestra la relación entre el nombre de cada circuito y el que debe asignarse a la carpeta contenedora para el PdC construido:

Circuito	Nombre de la carpeta	Nombre del fichero de configuración	
		XILINX	ALTERA
AES decoder	aes_dec	aes_dec.prj	aes_dec.qsf
AES encoder	aes_enc	aes_enc.prj	aes_enc.qsf
AES128	aes128_fast	aes128_fast.prj	aes128_fast.qsf
Cordic Core	sc_corproc	sc_corproc.prj	sc_corproc.qsf
CRC ultimate	ucrc_par	ucrc_par.prj	ucrc_par.qsf
DDS_Synthesizer	dds_synthesizer	dds_synthesizer.prj	dds_synthesizer.qsf
FPU	fpu	fpu.prj	fpu.qsf
Hilbert transform	analytic_filter_h_a1	analytic_filter_h_a1.prj	analytic_filter_h_a1.qsf
I2CSlave	i2cSlaveTop	i2cSlaveTop.prj	i2cSlaveTop.qsf
Reed Solomon decoder	rs_dec	rs_dec.prj	rs_dec.qsf
Reed Solomon encoder	rs_encode	rs_encode.prj	rs_encode.qsf
SD/MMC bootloader	spi_boot	spi_boot.prj	spi_boot.qsf
Sinuosidal computer	sin	sin.prj	sin.qsf
SPI Master Receiver for ADC	adcRecv	adcRecv.prj	adcRecv.qsf
T400 uC	t400_core	t400_core.prj	t400_core.qsf
UART16750	uart_16750	uart_16750.prj	uart_16750.qsf

Tabla 41. Nombre de cada IP del PdC

Es importante mantener una homogeneidad entre el nombre de la carpeta contenedora de los ficheros del circuito, de la entidad de mayor jerarquía y el del fichero de configuración para cada herramienta (prj si es ISE ó qsf si se trata de Quartus II). Este nombre debe ser el que tenga asignada la entidad superior del circuito. Si no es así, algunas de las herramientas producirán errores.

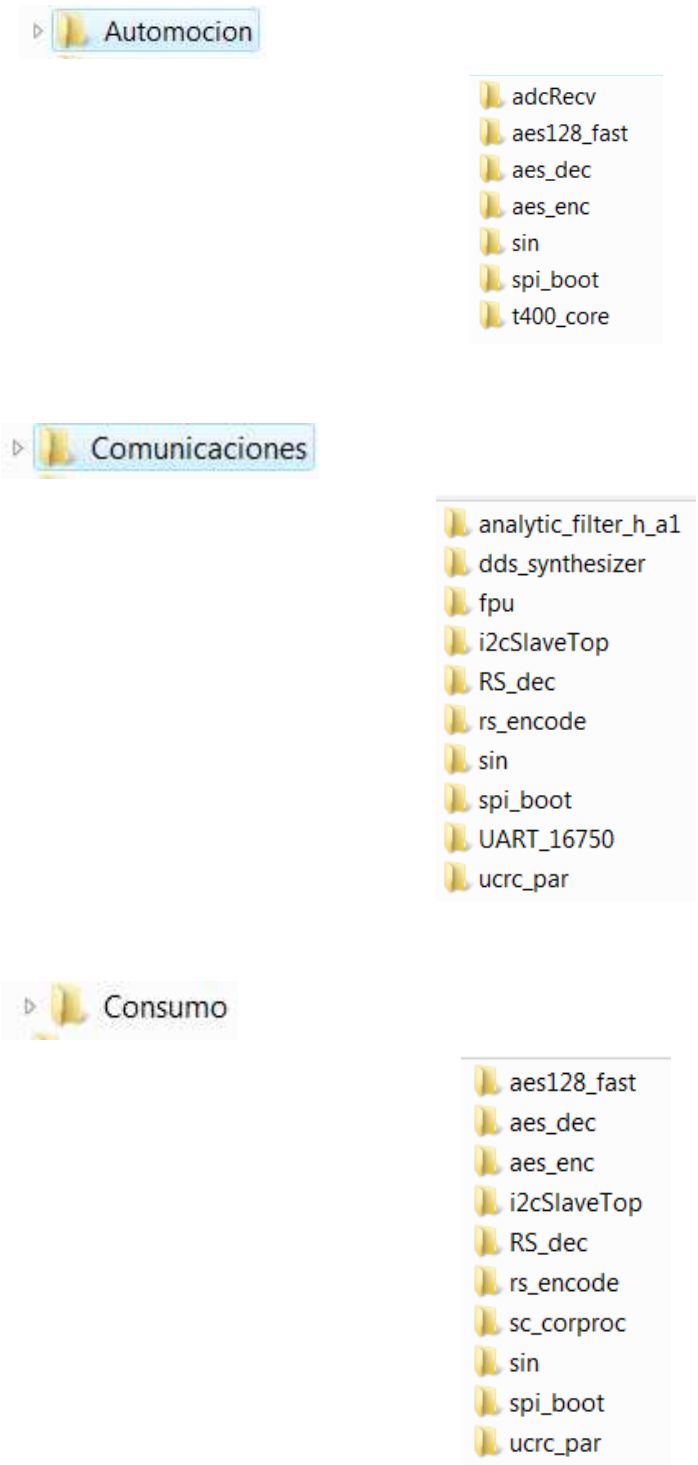


Figura 39. Aspecto final de las carpetas creadas para alojar el PdC

En la Figura 39 aparece la imagen de la organización de carpetas a construir para albergar cada uno de los IPs de cada

PdC según sector. La misma organización se establecerá por cada fabricante, en este caso Xilinx y Altera.

Paso 3

Mediante un editor de textos, escribir los ficheros de configuración de cada herramienta, y ubicarlos en la carpeta del circuito al cual correspondan.

*a) Fichero de configuración de Xilinx: *.PRJ*

Un fichero *<nombre_entidad_top>.prj* a incluir en cada carpeta de cada circuito del PdC. Contendrá tantas líneas como ficheros HDL necesite para completar su diseño. Cada línea incluirá información sobre el lenguaje en que está descrito el módulo, librería donde se debe compilar y su nombre, incluyendo extensión. Se reserva la última línea para el fichero de mayor jerarquía.

<lenguaje> <librería de compilación> "<nombre del fichero>" [14]

*a) Fichero de configuración de Altera: *.QSF*

Un fichero *<nombre_entidad_top>* QSF por cada carpeta de cada circuito seleccionado para el PdC. Incluye una línea por cada fichero necesario para el proyecto, otra para indicar el nombre de la entidad del bloque de mayor jerarquía. Se añadirán algunas opciones necesarias para un correcto análisis de potencia.

set_global_assignment -name TOP_LEVEL_ENTITY <nombre de la entidad top>
set_global_assignment -name VERILOG_FILE "<nombre del fichero>"[15]

[14] Las comillas son estrictamente necesarias en el caso de que el nombre incluya caracteres especiales como guión o espacio entre otros.

set_global_assignment -name POWER_PRESET_COOLING_SOLUTION "NO HEAT SINK WITH STILL AIR"[15]
set_global_assignment -name POWER_BOARD_THERMAL_MODEL "NONE (CONSERVATIVE)"[16]

Paso 4

Copiar los ficheros VHDL o verilog (*.vhdl o *.v) desde la ubicación donde se realizó la descarga de los proyectos-IPs de Opencores hacia estas nuevas carpetas, de modo que cada una incluya el fichero de configuración más los archivos HDL Figura 40 .

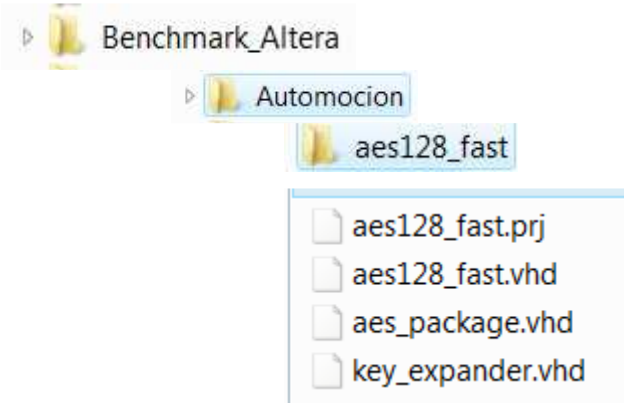


Figura 40. Jerarquía de ficheros por cada IP del PdC

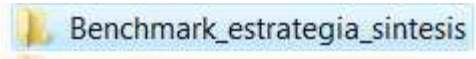
Paso 5

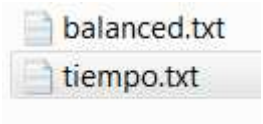
En la misma ruta, disco duro c, que se crearon las carpetas Benchmark_Xilinx y Benchmark_Altera, añadir la carpeta de nombre "Benchmark_estrategia_sintesis" con los dos ficheros de texto que incluye. Estos ficheros recogen todos los parámetros necesarios para que tenga lugar la fase de síntesis con las mismas condiciones que con el interfaz gráfico de ISE WebPack,

[15] Esta sentencia establece inexistencia de disipador y un flujo de aire de 0 FPM

[16] Si no se incluye esta línea la herramienta POWER de Quartus II considera que la placa tiene una resistencia térmica distinta de cero. De este modo no se considera el modelo de disipación placa-unión.

en adelante GUI, para cada una de las dos estrategias. Estos ficheros pueden consultarse en el Anexo A y reciben los nombres *balanced.txt* y *tiempo.txt*. Los ficheros de opciones de síntesis solo son válidos para la herramienta ISE de Xilinx.





4.4.2. Ficheros por lotes que ejecutan el PdC

Para automatizar las tareas de ejecución de los PdC, esto es, pasar por todas las etapas de implementación y análisis en cada una de las herramientas consideradas por cada IP y tecnología, se han desarrollado una serie de ficheros de ejecución por lotes para el intérprete de comandos de Microsoft Windows Vista. Estos archivos lanzarán, en modo desatendido, todos los procesos necesarios devolviendo a su término una serie de informes con los resultados y datos de interés.

Estos ficheros empiezan su labor generando una jerarquía de carpetas para alojar cada proyecto a resolver según sector y estrategia. Ver Figura 41,Figura 42 y Figura 43 .

TAREA 1 DEL FICHERO POR LOTES

PRIMER NIVEL

Crea carpetas por sector y estrategia de trabajo

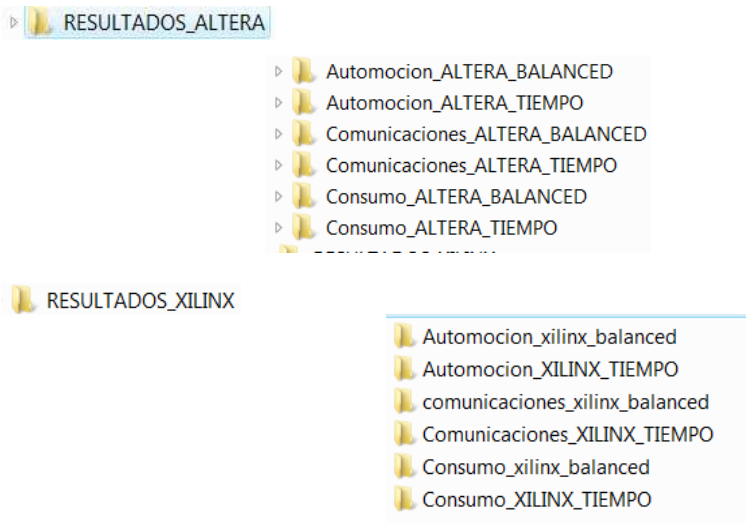
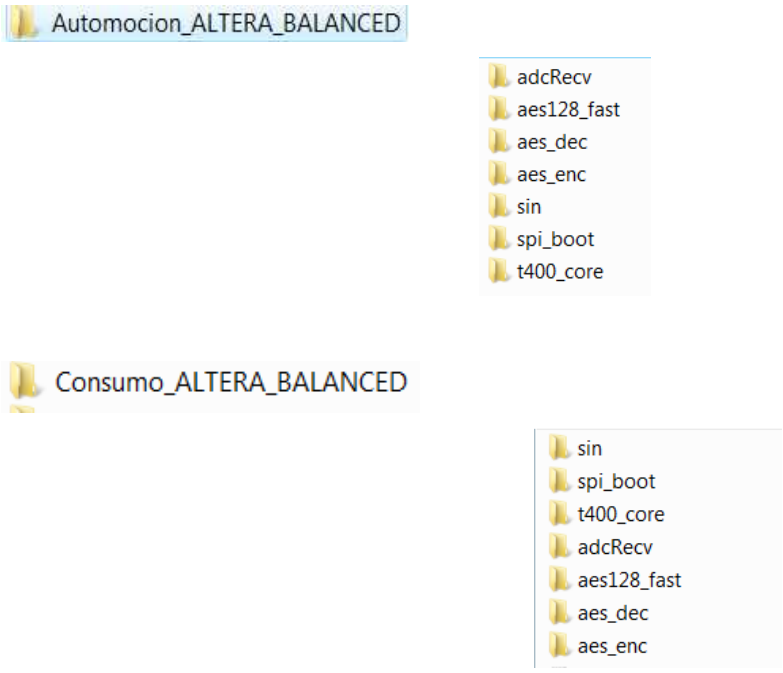


Figura 41. Organización de primer nivel del PdC

SEGUNDO NIVEL

Crea en las carpetas del nivel anterior, otras nuevas por cada IP del PdC. En la Figura 42 se muestran las correspondientes a la estrategia *Balanced*, siendo similar el resultado para la otra estrategia.



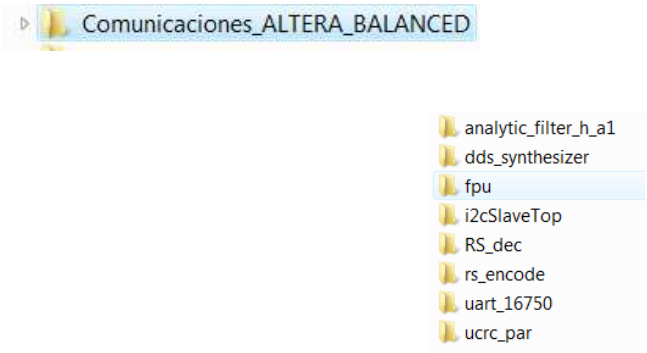


Figura 42. Organización del segundo nivel de jerarquía que guarda los resultados del PdC

TERCER NIVEL

El archivo por lotes crea una carpeta por cada tecnología sobre la que se vaya a implementar cada circuito. Para Altera se generará un total de 81 carpetas de tecnologías por cada circuito del PdC. En la Figura 43 se muestran solo algunas de ellas, no el total. Para Xilinx solo 106 carpetas-tecnologías y no las 165 seleccionadas debido al hecho de que todo el proceso de implementación y análisis no difiere para los modelos industriales y comerciales, Figura 44.

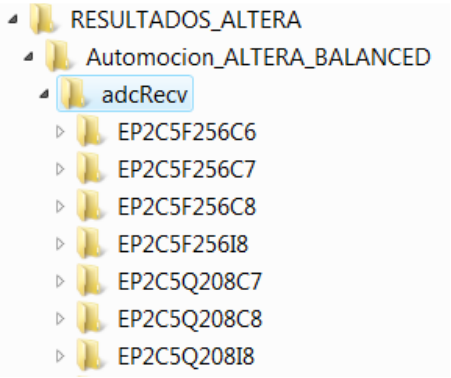


Figura 43. Imagen de la jerarquía completa de carpetas de resultados de Altera

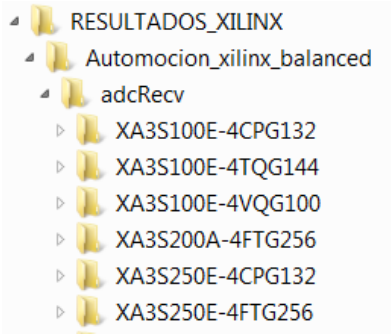


Figura 44. Imagen de la jerarquía completa de carpetas de resultados de Xilinx

TAREA 2 DEL FICHERO POR LOTES

Copia el conjunto de ficheros que compone cada proyecto-IP del PdC desde las carpetas donde se almacenaron según el apartado *Preparación del PdC*, hasta cada una de las recién creadas. El fichero bat correspondiente a la herramienta de Xilinx, también copia las opciones de síntesis en esas misma carpetas.

De este modo durante la ejecución cada herramienta dispondrá de todos los ficheros necesarios localmente: descripciones HDL de cada proyecto-IP, el fichero de configuración QSF de Quartus II, el fichero proyecto PRJ y las opciones de síntesis de las estrategias *tiempo y balanced*, estos últimos para ISE.

TAREA 3 DEL FICHERO POR LOTES

Para la herramienta de Xilinx, es necesario disponer de un fichero XST con toda la información para ejecutar la síntesis. El fichero bat correspondiente lo crea y lo añade a la carpeta correspondiente a cada tecnología (ver anexo A). Tras iniciar el proceso de sintesís con el fichero XST, continúa avanzando por cada fase del flujo de tareas de cada herramienta hasta completarlas todas, llamando a los comandos oportunos según herramientas, Tabla 42.

Tarea	ISE	Quartus II
Síntesis	El comando Synth lanza XST	Quartus_map
Translate	NGDBuild	
Mapping	MAP	
Place and Route	PAR	Quartus_fit
STA (Static Time Analysis)	TRACE	Quartus_tan
Ensamblado		Quartus_asm
Análisis de potencia	Xpwr	Quartus_pwr

Tabla 42. Comandos incluidos en los scripts según el entorno empleado

Cada comando se acompaña de una serie de valores para ajustar las herramientas tal y como trabajan en el GUI, con las estrategias: *balanceada* y *tiempo*. En el apartado siguiente se exponen los comandos de cada herramienta pormenorizadamente.

En la Tabla 43 se resume el nombre y los parámetros asociados a los ficheros por lotes (*.bat) desarrollados para la automatización de la ejecución del PdC y cuyo detalle se encuentra disponible en el Anexo A.1.

Nombre	Parámetros de usuario	Valores posibles
Ejecución_bench_xilinx.bat	%1	Balanced Tiempo
Ejecución_bench_altera.bat	%1	Balanced Tiempo

Tabla 43. Ficheros por lotes para la ejecución del PdC

Estos dos archivos por lotes incluyen los 16 circuitos IP del PdC y llamarán a otro fichero *bat* que a su vez se encarga de ejecutar todo el proceso por cada una de las 186 tecnologías objetivo de la implementación.

Nombre del BAT principal	Nombre del BAT llamado según parámetro %1
Ejecución_bench_xilinx.bat	Xilinx_balanced.bat
	Xilinx_tiempo.bat
Ejecución_bench_altera.bat	Altera_balanced.bat
	Altera_tiempo.bat

Tabla 44. Jerarquía de ficheros por lotes (*.bat)

En caso de querer incluir nuevos IPs en el PdC o nuevo modelos de FPGAs es fácil reutilizar estos ficheros, bastará con editarlos y reemplazar los circuitos y tecnologías actuales por las que se quieran utilizar.

INFORMES RESULTANTES

Tras lanzar los ficheros por lotes, se obtendrá por cada circuito y tecnología los siguientes informes de resultados:

Fase	Tipo de fichero (ISEWebpack / Quartus II)	
Síntesis	*.srp	*.map.summary
Traducción	*.ngr	*.fit.summary
Mapeado	*.map	
Emplazamiento e interconexionado	*.par	
Análisis estático de tiempo	*.twr	*.tao *.tan.summary
Análisis de Potencia	*.pwr	*.pow.summary

Tabla 45. Tipos de ficheros obtenidos al término de la ejecución del PdC

4.4.3. Comandos de cada herramienta

a) Webpack ISE 10 de Xilinx

En el caso de la herramienta de Xilinx, inicialmente se optó por hacer uso del comando XFLOW dado que incluye las etapas de síntesis, implementación y análisis estático de tiempo. XFLOW lee un fichero de entrada, un fichero que especifica el flujo de trabajo y un fichero que incluye las opciones con que se ejecutarán todos los programas.

Su sintaxis es:

xflow [–p partname][flow type][option file[.opt]] [xflow options] design_name

[–p partname]: *hace referencia a la tecnología empleada incluyendo encapsulado y grado de velocidad.*

[flow type]: *especifica la secuencia de tareas que quieren ejecutarse*

[option file[.opt]]: *estrategias de implementación*

[xflow options]: *diversas opciones como tipo de fichero de salida, etc*

La entrada a este comando es un diseño especificado como proyecto, *.prj, que incluye todos los ficheros HDL necesarios, siendo el ultimo citado el de mayor jerarquía en el diseño. Especificando los modos *–synth* e *–implement* tiene lugar la ejecución de los procesos XST, NGDBuild, MAP, PAR y TRACE (ver Tabla 42)

Las opciones posibles para el modo SYNTH son:

- xst_<lenguaje>.opt , donde lenguaje tomará los valores VHDL o verilog

y para IMPLEMENT:

- balanced: establece el valor de los parámetros para conseguir un equilibrio entre rendimiento y tiempo de CPU.
- -wd que indicará la carpeta de trabajo
- <…>.prj que indica ruta y nombre del proyecto que se implementará

Aunque este camino, a priori, ofrecía mayor claridad, requería establecer una separación de los circuitos del benchmark según si estaban descritos en VHDL o en Verilog. Por otro lado, aunque la información adjunta dejaba claro que PRJ era un tipo de fichero válido para ambos lenguajes, se comprobó que con descripciones Verilog, se producían errores. La resolución de este inconveniente pasaba por hacer nuevamente distinciones entre los ficheros de entradas según lenguajes, complicándose la claridad de la automatización. Por todo ello, se descarta el uso de este comando en los ficheros por lotes optándose por los específicos de cada tarea.

A continuación se relacionan los comandos y parámetros empleados en los ficheros *xilinx_balanced.bat y xilinx_tiempo.bat*:

Nombre del comando / Fase	Entrada/Salida	Parámetros utilizados (Balanced/tiempo)	Significado del parámetro (Balanced/tiempo)
XST (run)[17] / Síntesis	XST/NGC	Opciones de síntesis (txt) (ficheros txt)	
NGBUILD/Traducción	NGC/NGD	-nt timestamp	Actualiza los ficheros conforme a la fecha de creación
		-i	Ignora el fichero de restricciones UCF si lo hay
		-p	Indica la tecnología objetivo
MAP/Mapeado	NGD/NCD	-cm area	Trata de reducir el número de LUTs
		-pr off	Evita utilizar FF de E/S
		-k 4	Tamaño máximo de función manejado
		-c 100	Utiliza todos los CLB disponibles y reduce el nivel de empaquetado
		- p	Tecnología objetivo
		-logic_opt off	Evita repetir el emplazamiento
		-xe n	Nivel normal de extraesfuerzo
		-ol high	Nivel normal de esfuerzo
		-timing	Activa el mapeado timing-driven
		-t 1	Valor inicial de la tabla de costes

[17] El fichero XST establece la carpeta creada por tecnología y circuito como temporal y como ruta de la librería de compilación work. Las opciones de síntesis se copian desde el fichero TXT que las contiene.

Nombre del comando / Fase	Entrada/Salida	Parámetros utilizados (Balanced/tiempo)		Significado del parámetro (Balanced/tiempo)	
			-power off		Optimiza sin consideraciones de consumo
			-cm area		Reduce el número de LUTs
PAR / Emplazamiento e interconexionado	NCD/NCD	-ol std		Nivel de esfuerzo de P&R básico	
		-t 1		Valor inicial de la tabla de costes	
		-w		Sobreescribe los ficheros NCD existentes	
			-rl high		Máximo esfuerzo en el ruteado
			-pl high		Máximo esfuerzo en el emplazamiento
			-xe n		Establece el nivel de extra-esfuerzo de la herramienta
TRACE / Análisis estático de tiempo	NCD/TWR				
XPWR[18] / Análisis de potencia	NCD/PWR				

Tabla 46. Comandos de ISE WebPack incluidos en el fichero de automatización del PdC

[18] Por defecto emplea para el cálculo una frecuencia de 12,5MHz, un toggle-rate del 12,5%, una carga capacitiva por salida de 5pF, flujo de aire 0ºC, 25ºC de temperatura ambiente y ausencia de disipador térmico.

b) Comandos de Quartus II

Son tres los ficheros bat construidos para automatizar la ejecución del PdC en el entorno Quartus II 9.0 de Altera. Los dos primeros se diseñan ajustando los parámetros de los diferentes comandos a las estrategias denominadas *balanced* y *speed* (aquí denominada tiempo) según el GUI. El tercer y último efectúa llamadas a los anteriores en función de la estrategia especificada como parámetro.

Nombre del comando / Fase	Parámetros (Balanced / Speed)		Significado del parámetro
Quartus_map / Análisis, síntesis y mapeado	--part		Tecnología objetivo
	--optimize= balanced	--optimize= speed	Técnica de optimización
Quartus_fit / Emplazamiento e interconexionado	--part		Tecnología objetivo
	--effort= auto	--effort= standard	Esfuerzo del la técnica
Quartus_asm / ensamblado[19]			
Quartus_tan / Análisis estático de tiempo	--tao		Herramienta Classic Timing Analyzer, genera un fichero de salida en formato ASCII.
Quartus_pow / Análisis de potencia[20]	--no_input_file		No utiliza vector de estímulos
	--default_input_io_toggle_rate =1562500transitions/s		1.562.500[21] transiciones por segundo para E/S
	--use_vectorless_estimation=off		S vector de estimación

[19] Genera el fichero de programación, en Quartus II es imprescindible pasar por esta fase para obtener un análisis de potencia

[20] No se utilizará vector de estímulos. Esta situación da lugar a que el resultado no puede considerarse demasiado fiable como valor absoluto.

[21] Se considera la misma frecuencia de conmutación que utiliza por defecto Webpack ISE

Nombre del comando / Fase	Parámetros (Balanced / Speed)	Significado del parámetro
	--default_toggle_rate =1562500transitions/s)	1.562.500 transiciones por segundo para las salidas de todos los registros

Tabla 47. Comados de Quartus II incluidos en el fichero de automatización del PdC

4.5. Consideraciones

En tono de recapitulación en este capítulo se ha expuesto una metodología que permite extraer un conjunto de proyectos-IP del repositorio público Opencores, válidos para conformar un patrón de comparación de FPGAs.

Siguiendo el método se ha establecido un PdC compuesto por 16 IPs que cumplen los requisitos adecuados para que las herramientas de diseño de Xilinx y Altera los traten e interpreten en igualdad de condiciones sin tener que modificar sus configuraciones de base.

Se ha descrito un procedimiento de actuación previa al uso de una serie de archivos de ejecución por lotes, que han sido desarrollados durante este trabajo al objeto de automatizar la implementación y análisis de cada uno de los IPs del PdC sobre todas las tecnologías y las dos herramientas establecidas. El inicio de los mismos se hará desde la línea de comandos de la consola de Windows Vista utilizando un equipo PC Fujitsu-Siemens QuadCore con 4GB de memoria RAM.

Se describen los comandos y parámetros incluidos en los ficheros por lotes de cara a homogeneizar su comportamiento con el

obtenido caso de haber trabajado con el interfaz gráfico de cada herramienta.

Al término de este capítulo los objetivos c),d) y f.1) marcados al inicio han quedado resueltos y se dispone de un total de 15.960 ficheros que detallan el éxito o fracaso de cada implementación y análisis así como los datos de lógica utilizada o de velocidad. Toda esta información constituye los datos experimentales necesarios para el siguiente capítulo donde se establecerá un sistema de puntuación y clasificación de FPGAs.

5. Métrica y puntuación

Tal y como se ha comentado en capítulos anteriores definir una métrica genérica que se ajuste a todos los criterios que un fabricante pueda esgrimir a la hora de decantarse por un modelo de FPGA u otro para su diseño, es una tarea casi imposible. Algunos productos requerirán frecuencias de trabajo muy altas, otros un gran número de entradas y salidas, otros muchos recursos lógicos, y la lista podría continuar.

Sin embargo, algunas magnitudes son referencias clásicas en el mundo de las FPGAs y sobre ellas se hará un breve repaso, a la vez que se analiza el método de cálculo que emplea cada fabricante a la hora de presentar esos valores. En último término se realizarán las correciones oportunas para presentar los parámetros, de modo que la información que faciliten tenga idéntico significado, independientemente de la herramienta de la cual procedan.

5.1. Magnitudes clásicas

5.1.1. Parámetros temporales

Las herramientas de análisis de tiempo, en adelante STA, *Static Time Analysis*, realizan estimaciones temporales tras la fase de síntesis o bien tras la implementación e interconexionado de los recursos participantes en el diseño. El valor más próximo a la realidad es el que se obtiene en el segundo caso. Se opta por tanto por analizar los datos devueltos tras este análisis.

Los circuitos integrantes del PdC no incluyen estructuras complejas de reloj ni más de un dominio, razón por la cual los resultados obtenidos mediante los STA de las distintas herramientas serán comparables, con alguna pequeña salvedad. Tampoco es necesario modificar ningún elemento de su configuración ni añadir restricciones al diseño.

El formato empleado en cada STA obligará a realizar una lectura diferente de los informes de cada fabricante para devolver así resultados equiparables.

5.1.1.1 Frecuencia máxima

Un parámetro clásico a la hora de evaluar el rendimiento de un circuito, es la máxima frecuencia a la que éste es capaz de trabajar. La máxima frecuencia viene determinada por el camino crítico, esto es, el máximo retardo que se produce entre dos registros del circuito, Figura 45.

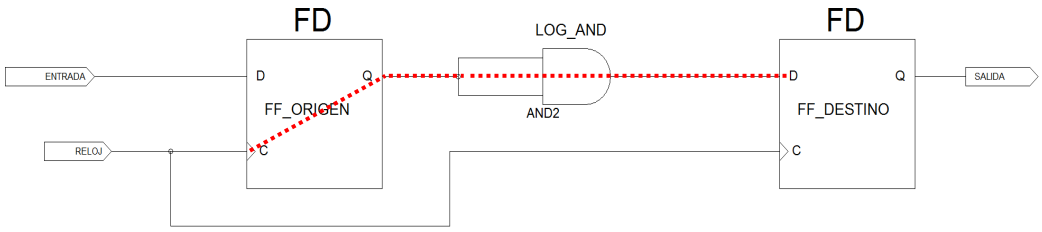


Figura 45. Ejemplo de retardo máximo entre dos flip-flops

La (Ecuación 8) define esta magnitud y será la utilizada en este trabajo.

$$T_{FF\text{-}origen\ FF\text{-}destino} = Clock\text{-}to\text{-}out_{FF\text{-}ORIGEN} + Retardo\ combinacional\ intermedio + Retardo\ conexiones + Tiempo\ establecimiento_{FF\text{-}destino}$$

(Ecuación 8)

5.1.1.2 Tiempo de establecimiento externo (Setup)

Tiempo de antelación del dato en el pad de entrada respecto al reloj del circuito para no violar el tiempo de establecimiento interno del flip-flop, Figura 46.

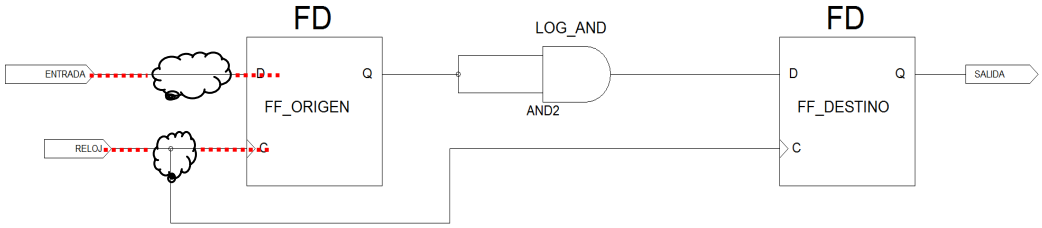


Figura 46. Representación del tiempo de establecimiento externo del FF de entrada

$$T_{establecimiento\ externo\ minimo} = Retardo\ del\ dato_{max}\ (lógico\ y\ conexiones) + Tiempo\ de\ establecimiento\ interno\ del\ FF - Retardo\ del\ reloj_{min}\ (lógico\ y\ conexiones)$$

(Ecuación 9)

5.1.1.3 Tiempo de mantenimiento externo (Hold)

Marca el tiempo que el dato debe permanecer estable en el flip-flop de entrada tras la llegada del flanco activo de reloj, para no violar el tiempo de mantenimiento interno del flip-flop.

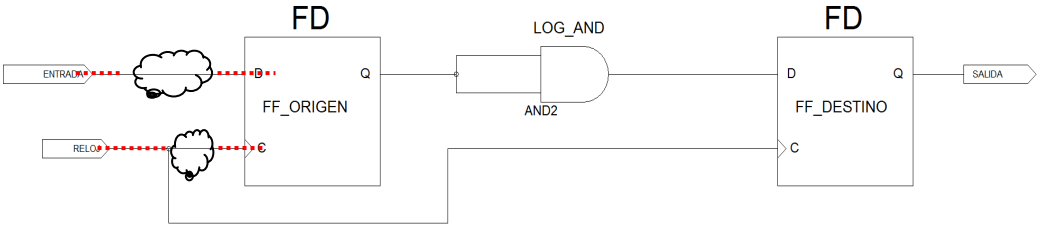


Figura 47. Hold externo del circuito

$$T_{mantenimiento\ externo\ máximo} = Retardo\ del\ reloj_{máx}\ (lógica\ y\ conexiones) + Tiempo\ de\ establecimiento\ interno\ del\ FF - Retardo\ del\ dato_{min}\ (lógica\ y\ conexiones)$$

(Ecuación 10)

5.1.1.4 Tiempo de propagación desde el último flip-flop al terminal de salida (Clock-to-pad)

Tiempo máximo que transcurre desde la entrada de la señal de reloj al flip-flop de salida, hasta que se dispone de un valor estable en el pad.

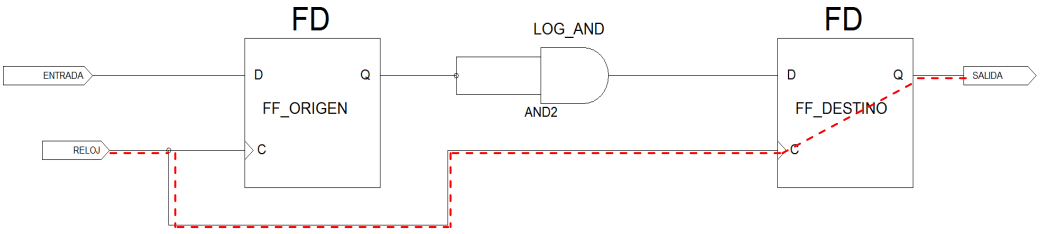


Figura 48. Tiempo máximo que transcurre hasta disponer de un valor estable de salida

$$T_{salida\ a\ pad\ máx} = Retardo\ del\ reloj_{máx}\ (lógica\ y\ conexiones) + Clock\text{-}to\text{-}out_{máx}\ interno\ del\ FF + Retardo\ del\ dato\ de\ salida_{máx}\ (lógica\ y\ conexiones)$$

(Ecuación 11)

Como se comentó más arriba, cada herramienta devuelve los parámetros temporales citados atendiendo a diferentes criterios y no siguen las ecuaciones expuestas estrictamente. En la Tabla 48 se recoge la denominación que reciben estas magnitudes en cada una.

Herramienta	ISE 10.1.03	Quartus II 9.0
Analizador de tiempo	Timing Analyzer	Classic Timing Analisys
Informes que contienen los datos temporales	*.twr	*.tao *.tan.rpt
Tiempo$_{mx}$ del camino crítico (ns)	Clock to setup on destination <nombre_reloj>	*Actual Longest P2P Time* + µTco(FF origen)+ µTsu (FF destino)[22]
Tiempo establecimiento (ns)	Setup to Clock <nombre_reloj>	Tsu
Tiempo de mantenimiento (ns)	Hold to clock <nombre_reloj>	Th
Tiempo hasta el pad de salida (ns)	Clock <nombre_reloj> to Pad	Tco

Tabla 48. Denominación de los tiempos citados en cada herramienta

Las Figura 49, Figura 50 y Figura 51 muestran un ejemplo de los informes obtenidos tras realizar el análisis de tiempo con ISEWebpack y con Quartus II, se destaca en sombreado los parámetros referidos.

```
Release 10.1.03 Trace  (nt)

Copyright (c) 1995-2008 Xilinx, Inc.  All rights reserved.

C:\Xilinx\10.1\ISE\bin\nt\unwrapped\trce.exe

c:\Automocion_xilinx_balanced\adcRecv\XA3S50-4PQG208\adcRecv.ncd -o

c:\Automocion_xilinx_balanced\adcRecv\XA3S50-4PQG208\adcRecv.twr
```

[22] El STA que se utiliza en Quartus II es Classic Timing Analyzer, éste no devuelve de forma directa el retardo máximo entre flip-flops, por lo que se calculará usando esta expresión cuyos sumandos se encuentran en los informe *.tao y *.tan.rpt

```
Design file:              adcRecv.ncd

Physical constraint file: adcRecv.pcf

Device,package,speed:     xa3s50,pqg208,-4 (PRODUCTION 1.39 2008-01-09)

Report level:             summary report, limited to 0 items per constraint

[…]

Data Sheet report:
-----------------
All values displayed in nanoseconds (ns)
```

Setup/Hold to clock CLK

Source	Setup to clk (edge)	Hold to clk (edge)	Internal Clock(s)	Clock Phase
ContinueStart	2.792(R)	-1.095(R)	Shift_OBUF	0.000
	1.758(F)	-0.305(F)	Shift_OBUF	0.000
SDI	0.537(R)	0.662(R)	Shift_OBUF	0.000
ShutDown	3.868(R)	-1.955(R)	Shift_OBUF	0.000
	2.834(F)	-1.165(F)	Shift_OBUF	0.000
Start	3.313(R)	-0.085(R)	Shift_OBUF	0.000

Clock CLK to Pad

Destination	clk (edge) to PAD	Internal Clock(s)	Clock Phase
DQ<0>	8.814(R)	Shift_OBUF	0.000
DQ<1>	8.856(R)	Shift_OBUF	0.000
DQ<2>	8.089(R)	Shift_OBUF	0.000
DQ<3>	8.821(R)	Shift_OBUF	0.000
DQ<4>	8.089(R)	Shift_OBUF	0.000
DQ<5>	8.085(R)	Shift_OBUF	0.000
DQ<6>	8.060(R)	Shift_OBUF	0.000
DQ<7>	8.056(R)	Shift_OBUF	0.000
DQ<8>	8.057(R)	Shift_OBUF	0.000
DQ<9>	8.043(R)	Shift_OBUF	0.000
DQ<10>	8.083(R)	Shift_OBUF	0.000
DQ<11>	8.086(R)	Shift_OBUF	0.000
DQ<12>	8.105(R)	Shift_OBUF	0.000
DQ<13>	8.804(R)	Shift_OBUF	0.000
DQ<14>	8.157(R)	Shift_OBUF	0.000
DQ<15>	8.091(R)	Shift_OBUF	0.000
Ready	8.568(F)	Shift_OBUF	0.000
SCK	10.261(R)	Shift_OBUF	0.000
nSS	10.228(R)	Shift_OBUF	0.000
	10.043(F)	Shift_OBUF	0.000

Clock to Setup on destination clock CLK

Source Clock	Src:Rise Dest:Rise	Src:Fall Dest:Rise	Src:Rise Dest:Fall	Src:Fall Dest:Fall
CLK	3.770			

```
Analysis completed Fri Jul 24 17:27:48 2009
-------------------------------------------------------------------------------

Trace Settings:
-------------------------
Trace Settings

Peak Memory Usage: 89 MB
```

Figura 49. Informe TWR generado por Timing Analyzer (ISE 10)

```
Date: 06/20/2009 01:56:41
Analysis Type: all
Compiler Settings: fm
Device: EP2C5F256C8

--------------------------------------------------------------------------------
Clock Setup: 'CLK'
--------------------------------------------------------------------------------
Path Number                  : 1
Slack                        : N/A
Actual fmax (period)         : 43.27 MHz ( period = 23.113 ns )
From                                                                          :
nco:mynco|rom:myrom|altsyncram:Mux0_rtl_0|altsyncram_ufu:auto_generated|ram_block1a0~por
ta_address_reg0
To                           : phase_detector:myphase|phase_output[7]
From Clock                   : CLK
To Clock                     : CLK
Required Setup Relationship  : None
Required Longest P2P Time    : None
Actual Longest P2P Time      : 22.788 ns

Path Number                  : 2
Slack                        : N/A
Actual fmax (period)         : 43.27 MHz ( period = 23.113 ns )
From                                                                          :
nco:mynco|rom:myrom|altsyncram:Mux0_rtl_0|altsyncram_ufu:auto_generated|ram_block1a0~por
ta_address_reg1
To                           : phase_detector:myphase|phase_output[7]
From Clock                   : CLK
To Clock                     : CLK
Required Setup Relationship  : None
Required Longest P2P Time    : None
Actual Longest P2P Time      : 22.788 ns

--------------------------------------------------------------------------------
tsu
--------------------------------------------------------------------------------
Path Number   : 1
Slack         : N/A
Required tsu  : None
Actual tsu    : 22.224 ns
From          : FMIN[0]
To            : phase_detector:myphase|phase_output[7]
To Clock      : CLK

Path Number   : 2
Slack         : N/A
Required tsu  : None
Actual tsu    : 22.151 ns
From          : FMIN[1]
To            : phase_detector:myphase|phase_output[7]
To Clock      : CLK


--------------------------------------------------------------------------------
tco
--------------------------------------------------------------------------------
Path Number   : 1
Slack         : N/A
Required tco  : None
Actual tco    : 9.699 ns
From          : fir:myfir|dmout[11]
To            : DMOUT[11]
From Clock    : CLK

Path Number   : 2
Slack         : N/A
Required tco  : None
Actual tco    : 9.620 ns
From          : fir:myfir|dmout[4]
To            : DMOUT[4]
From Clock    : CLK


--------------------------------------------------------------------------------
```

```
th
-----------------------------------------------------------------------------------------
Path Number    : 1
Minimum Slack  : N/A
Required th    : None
Actual th      : -1.410 ns
From           : FMIN[3]
To             : phase_detector:myphase|phase_output[2]
To Clock       : CLK

Path Number    : 2
Minimum Slack  : N/A
Required th    : None
Actual th      : -1.491 ns
From           : FMIN[4]
To             : phase_detector:myphase|phase_output[4]
To Clock       : CLK
```

Figura 50. Informe TAO generado por Classic Timing Analysis (Quartus II)

```
Classic Timing Analyzer report for fm
Sat Jun 20 01:56:41 2009
Quartus II Version 9.0 Build 132 02/25/2009 SJ Web Edition
---------------------
; Table of Contents ;
---------------------
  1. Legal Notice
  2. Timing Analyzer Summary
  3. Timing Analyzer Settings
  4. Clock Settings Summary
  5. Parallel Compilation
  6. Clock Setup: 'CLK'
  7. tsu
  8. tco
  9. th
 10. Timing Analyzer Messages
(…)

Info: + Micro clock to output delay of source is 0.260 ns
    Info: + Micro setup delay of destination is -0.040 ns

(…)
```

Figura 51. Informe TAN.RPT generado por Classic Timing Analysis (Quartus II)

Inicialmente se leerán todos los parámetros de tiempo expuestos por cada circuito y tecnología, más adelante se debatirá sobre la idoneidad de incluirlos o no en el sistema de puntuación que se establezca.

5.1.2. Área de ocupación

Otro parámetro clásico es el área total de recursos lógicos ocupada por el diseño. En los PdC precursores esta medida venía dada por el número de veces que un determinado circuito podía instanciarse sobre una misma tecnología. De este modo, establecían la comparación entre distintos modelos de FPGA, considerándose mejor el que era capaz de albergar más instancias.

Sin embargo, este trabajo defiende la idea de que puede establecerse una comparación justa instanciando cada circuito una sola vez y evaluando la ocupación porcentual de los recursos hardware del dispositivo objetivo. Al instanciar una sola vez, el núcleo se emplazará e interconectará en las mejores condiciones posibles, sucediendo lo mismo para todas las tecnologías que se utilicen como objetivo. Por tanto, todas las FPGAs darán lo mejor de sí, y en esta situación se las evaluará.

Los informes recogen la cantidad total disponible y la cantidad utilizada de unidades lógicas, los bits de bloques de memoria disponibles y utilizados, de multiplicadores existentes y utilizados y de terminales de entradas/salidas disponibles y utilizados.

5.1.2.1 Unidades lógicas

Los bloques lógicos básicos de las distintas arquitecturas analizadas son diferentes tal y como ya se detalló en el capítulo 3. A lo largo de este trabajo se hace referencia a ellos utilizando el término “unidad lógica” y se entiende como tal a la “unidad” en que los informes expresan la ocupación lógica de cada dispositivo y que está formado al menos por una

tabla de búsqueda (LUT) y un flip-flop. La unidad lógica en las tecnologías de Xilinx recibe el nombre de *slice* y de *logic element* en las de Altera.

Tal y como muestra la Tabla 49, presentan unidades lógicas que cuentan con un número diferente de LUTs y flip-flops, razón por la cual a la hora de expresar la ocupación conseguida con cada circuito del PdC se haya optado por hablar de valores relativos, en tanto por ciento, y no por absolutos.

El porcentaje de unidades lógicas expresa qué cantidad de unidades lógicas necesita un diseño para implementarse sobre una FPGA. Este mismo valor, informa de qué cantidad de unidades lógicas quedan disponibles en el dispositivo objetivo, tras haber ubicado en él un circuito del PdC.

De cara a la selección de un modelo de FPGA e incluso a la determinación de su puntuación, es interesante conocer el porcentaje de ocupación tanto para verificar la capacidad del dispositivo para alojar el diseño, como por conocer los recursos sobrantes. Disponer de lógica ociosa ofrece la posibilidad de implementar sobre ella circuitería adicional y conseguir un mayor nivel de integración del producto final.

Tras haber ejecutado el PdC, se presenta la oportunidad de revisar los resultados con el objetivo de comprobar si puede establecerse una relación entre las unidades lógicas de ambos tipos de arquitectura, comparando la cantidad empleada en cada caso por cada circuito. En el capítulo 7 se expondrá el análisis realizado y las conclusiones obtenidas.

5.1.2.2 Bloques de memoria

Los bloques de memoria son un recurso con cada vez mayor presencia en las tecnologías económicas. Al igual que sucede con las unidades lógicas, presentan diferencias entre fabricantes. Los informes de resultados expresan la utilización de memoria en diferentes unidades. Para homogeneizar la información se harán las conversiones oportunas para que el término memoria haga referencia al porcentaje de bits utilizados o sobrantes. Elegir un modelo de FPGA que se ajuste a un determinado diseño y mantenga libre los bloques de memoria puede ser una opción interesante que elimine la necesidad de utilizar algún circuito integrado externo para el producto final.

Las características de los bloques de memoria presentes en las FPGAs de Xilinx y Altera pueden verse en la Tabla 49.

5.1.2.3 Multiplicadores

La inclusión de bloque multiplicadores en las FPGA de bajo coste, ha sido uno de los agentes potenciadores de estas familias. Este elemento aritmético consigue grandes mejoras en el rendimiento de diseños orientados hacia el procesamiento digital de señal, uno de los campos más presentes en el sector de las comunicaciones y la electrónica de consumo.

Al igual que en el resto de recursos, cada fabricante los ofrece con algunas características diferentes, luego referir los resultados en términos relativos sigue siendo una buena solución. En esta ocasión el formato final expresara el porcentaje de multiplicadores de 18x18 bits utilizados y

sobrantes a partir de la información que proporcionan los informes.

La siguiente tabla recoge, por herramienta, las características y formato con que aparecen en los informes los recursos lógicos citados.

	ISE 10.1.03	Quartus II 9.0
Informe generado	<nombre>.par	<nombre>.fit.rpt ó <nombre>.flow
Unidades lógicas[23]	1 Slice = 2 LUTs + 2 Flip-flops	1 Logic Element (LE) = 1 LUT + 1 Flip-flop
	Total Slices , % Slices	Total LE, % LE
Entradas / Salidas	Total IOB, % IOB	Total pins, % pins
Bits de memoria	Síncronos configurables en puertos y capacidad. 1 bloque = 16Kbits (datos) + 2Kbits (paridad)=18Kbits	Síncronos configurables en puertos y capacidad. 1 bloque = 4Kbits (datos)+ 512 bits (paridad) = 4.608 bits
	Según modelo: Number of RAMB16, % Ram Number of RAMB16BWES	Total memory bits, % bits
Multiplicadores	Síncrono, asíncrono. Entradas de 18 bits y salida de 36 bits.	Síncrono, asíncrono. Configurables 9x9 ó 18x18 bits de entrada.
	Según modelo: Number of Mult18x18,% Mult Number of Mult18x18s,% Mult Number of Mult18x18SIOs, % Mult	Embedded Multiplier 9-bit elements, % elements Embedded Multiplier 18-bit elements, % elements

Tabla 49. Elementos lógicos según fabricantes: formato e identificación en los informes

[23] Se destacan solo las LUTs y FF, aunque cada unidad lógica cuenta con otros recursos como la lógica para la propagación del acarreo.

5.1.3. Potencia

El informe generado por el analizador de potencia, recoge los valores de potencia estática y dinámica así como la resistencia térmica entre distintos interfaces. Para este trabajo se realiza la lectura de la potencia total consumida en miliwatios.

Dado que el análisis realizado no es demasiado exhaustivo al no contar con valores reales de funcionamiento de cada núcleo implementado, sus resultados no serán válidos en términos absolutos aunque si como cifra relativa, esto es cuánto más consume una tecnología que otra. Esta medida será suficiente para establecer una comparación entre distintos modelos y en base a ella clasificarlas en orden de preferencia.

5.1.4. Tiempo de CPU y memoria

Es posible conocer el tiempo de CPU y la memoria de pico utilizados en las distintas etapas del proceso de implementación y análisis. La ubicación y existencia de esta información depende de cada fabricante. La herramienta Quartus II genera un informe, FLOW.RPT, que incluye el tiempo completo y de CPU invertido en cada tarea así como del pico de memoria requerido en cada una. Webpack ISE incluye esta información en el fichero generado al término de cada fase, ofrece el tiempo real y de CPU para la síntesis, mapeado y emplazamiento y ruteado. Para estas mismas herramientas y para el análisis de tiempo estático refiere el pico de memoria empleada.

El formato de tiempo empleado es diferente según fabricante e informe, por ello, habrá que realizar una conversión a segundos de los tiempos medidos. En cuanto a la memoria de pico, se leerá el dato de memoria máxima utilizada y se expresará en megabytes.

5.1.5. Precio por unidad

Las tecnologías usadas como objetivo en los ensayos de este trabajo son las más económicas de FPGAs de cada fabricante. Por otro lado, los circuitos patrón elegidos son representativos de sectores comerciales donde prima la idea de un precio reducido para el producto final. Estas razones llevan a pensar que la variable precio es tan interesante como las clásicas ya citadas: frecuencia, área y potencia.

Incluir este parámetro en la valoración del comportamiento de cada FPGA ante el PdC, es novedoso respecto a las propuestas anteriores donde casi siempre primaba la velocidad. Si frecuencia o potencia son parámetros poco relevantes en un diseño la variable precio puede ser la que marque la diferencia a la hora de decantarse por un modelo u otro.

Otra razón para incluir el precio en este sistema de puntuaciones es lo heterogéneo de los recursos y capacidades del lote de modelos revisados, esto es, el conjunto de FPGAs presenta distintos grados de velocidad, diferentes número de recursos lógicos y de terminales. Aquellos modelos con mejores prestaciones, serán también los más caros, por eso, al relacionar las magnitudes con el precio por unidad, se les “penalizará”, sin embargo, las tecnologías mas económicas se verán “beneficiadas” con esta corrección y el conjunto quedará más homogéneo.

El precio se incluye en una tabla de la base de datos que almacena toda la información de ejecución del PdC y se ha obtenido del catálogo de Digikey que distribuye FPGAs de Xilinx y Altera. Viene expresado en dólares y hace referencia a la compra de una sola unidad.

5.2. Método de puntuación y clasificación de FPGAs

Es difícil señalar algún producto cuya existencia en el mercado sea exclusiva. En general, siempre se dispone de una amplia oferta de modelos que podrían satisfacer unos determinados requisitos.

Por ejemplo, el caso de los ordenadores domésticos, la variedad es impresionante y elegir cuál comprar dependerá principalmente de la tarea a que se vaya a destinar y el dinero que se pretenda invertir en él. Partir de un precio límite quizás sea un buen comienzo para reducir el número inicial de opciones, luego se tratará de averiguar qué equipo da las mejores prestaciones por esa cantidad o menos.

Revisando revistas populares sobre informática, se pueden encontrar artículos que presentan comparativas entre PCs de distintas marcas y precios que puntúan sus características más relevantes como memoria, velocidad del procesador, capacidad, velocidad del disco duro, etc, y finalmente les asignan una determinada nota calculada como una relación calidad/precio.

Cuando se parte de la premisa de que un producto contará con una FPGA de bajo coste para su fabricación, es lógico pensar que el precio final del mismo es la característica más importante para su fabricante. La opción más interesante será aquella que al menor precio le ofrezca más: mejor velocidad, mayor cantidad de recursos disponibles de entradas/salidas y el menor consumo, en resumen: “más por menos”.

El sistema de puntuación que se propone nace al amparo de estas conjeturas y trata de clasificar las FPGAs de bajo coste

atendiendo al mejor resultado en el PdC con el menor precio posible.

5.2.1. Cálculo de puntuaciones por parámetro (Pk)

La ejecución del PdC ha devuelto una gran cantidad de información sobre cómo se ha implementado cada núcleo en las FPGAs de la muestra. En el apartado anterior se repasaron las magnitudes más importantes y se expusieron las diferencias de sus formas según herramienta y como homogeneizarlas para que coincidieran en significado.

A continuación se propone un sistema de puntuaciones que utiliza cinco de las magnitudes comentadas relacionándolas con el precio y en base a ellas calcula una puntuación global por circuito y una final ante el PdC completo o por sector comercial. A partir de la puntuación final se clasificará la muestra utilizada de FPGAs.

Frecuencia máxima de trabajo versus precio (P1)

Un mismo circuito que se implementa sobre una FPGA de mayor tamaño y mayor grado de velocidad devolverá el mejor tiempo de registro a registro y por ende la mayor frecuencia de trabajo, pero también será el modelo más caro.

La frecuencia calculada a partir del tiempo de propagación entre registros más desfavorable, puede no ser real, ya que no considera las limitaciones propias de cada tecnología, el estándar de salida que se utiliza o la pendiente de cambio marcada. Pero, del mismo modo se han implementado todos los IPs del PdC en todas las parejas FPGA-herramienta y unas dan mejores resultados que otras.

La primera puntuación, denominada P1, relaciona por cada FPGA, la frecuencia máxima conseguida por cada IP del PdC con su precio unitario de mercado.

$$\mathbf{P1}_{i,j} = \frac{F_{max}(T_i, C_j)}{Precio\,(T_i)} \quad \textbf{(Ecuación 12)}$$

Donde

$T_i \equiv$ cada modelo de FPGA ($1 \leq i \leq m$). Para este trabajo m=246.
$C_j \equiv$ cada IP del PdC ($1 \leq j \leq n$). Para este trabajo n=16.

La Tabla 50 ejemplifica las puntuaciones P1 obtenidas en todas las FPGAs y todos los núcleos del PdC.

		Circuitos			
		C_1	C_2	…	C_n
Tecnologías	T_1	$P1_{1,1}$	$P1_{1,2}$	…	$P1_{1,n}$
	T_2	$P1_{2,1}$	$P1_{2,2}$	…	$P1_{2,n}$
	…	…	…	…	…
	T_m	$P1_{m,1}$	$P1_{m,1}$	…	$P1_{m,n}$

Tabla 50. Puntuación P1 que relaciona Fmax con precio por unidad

Unidades lógicas sobrantes versus precio (P2)

La siguiente unidad de puntuación, P2, establece una relación entre los recursos lógicos (slices ó LEs) sobrantes y el precio que se paga por cada FPGA. Se pretende conocer de esta forma el porcentaje de unidades lógicas aún disponibles tras la implementación de cada circuito del PdC, dada la heterogeneidad constructiva de la muestra de FPGAs.

Este enfoque, debe entenderse de la siguiente manera, un determinado circuito, C_i, se implementa en varias FPGAs con

distintas capacidades, la ocupación del mismo será similar en todas, luego valorar el área ocupada no aporta valor. Sin embargo, conocer el porcentaje de unidades lógicas disponibles tras concluir la implementación de C_i, y enfrentarlo con el precio a pagar por ese dispositivo establecerá las diferencias entre ese y otro modelo de igual precio. A igualdad de precio, la adquisición más interesante será la que presente más unidades lógicas de sobra tras implementar el mismo circuito.

$$\mathbf{P2}_{i,j} = \frac{\%UL_{sobran}{}^{24}(T_i, C_j)}{Precio\ (T_i)} \quad \textbf{(Ecuación 13)}$$

Donde

T_i ≡ cada modelo de FPGA ($1 \leq i \leq m$). Para este trabajo m=246.
C_j ≡ cada IP del PdC ($1 \leq j \leq n$). Para este trabajo n=16.

	Circuitos			
	C_1	C_2	…	C_n
T_1	$P2_{1,1}$	$P2_{1,2}$	…	$P2_{1,n}$
T_2	$P2_{2,1}$	$P2_{2,2}$	…	$P2_{2,n}$
…	…	…	…	…
T_m	$P2_{m,1}$	$P2_{m,1}$	…	$P2_{m,n}$

Tabla 51.Puntuación P2 que relaciona %Unidades lógicas sobrantes con precio por unidad

Bits de bloques de memoria sobrantes versus precio (P3)

El razonamiento utilizado para P2 se sigue a la hora de puntuar el resto de recursos lógicos de las FPGAs de bajo coste. Este tratamiento persigue a su vez, evaluar el trabajo realizado por

[24] La lectura en los informes extrae información de las unidades lógicas utilizadas y totales de cada dispositivo FPGA. Igual sucede con los bits de memoria y los multiplicadores.

la estrategia de equilibrio de las herramientas, asignándose mayor puntuación al caso que aproveche y optimice mejor todos sus recursos sin intervención del usuario. Esto es, si la configuración asociada a esta estrategia es capaz de instanciar adecuadamente todos los recursos disponibles en la FPGA y no se limita a construir el circuito con los bloques lógicos básicos.

El cálculo de P3 queda indicado por la Ecuación 14

$$\mathbf{P3}_{i,j} = \frac{\%\boldsymbol{BITs}_{\boldsymbol{sobran}}(\boldsymbol{T}_i,\boldsymbol{C}_j)}{\boldsymbol{Precio}\,(\boldsymbol{T}_i)} \quad \textbf{(Ecuación 14)}$$

Donde

T_i ≡ cada modelo de FPGA ($1 \leq i \leq m$). Para este trabajo m=246.
C_j ≡ cada IP del PdC ($1 \leq j \leq n$). Para este trabajo n=16.

	Circuitos			
	C_1	C_2	…	C_n
T_1	$P3_{1,1}$	$P3_{1,2}$	…	$P3_{1,n}$
T_2	$P3_{2,1}$	$P3_{2,2}$	…	$P3_{2,n}$
…	…	…	…	…
T_m	$P3_{m,1}$	$P3_{m,1}$	…	$P3_{m,n}$

Tabla 52.Puntuación P3 que relaciona %bits de bloques de memoria sobrantes con precio por unidad

Multiplicadores 18x18 sobrantes versus precio (P4)

La puntuación P4 sigue la línea de P2 y P3 y se calcula según

$$\mathbf{P4}_{i,j} = \frac{\%\boldsymbol{Multiplicadores}_{\boldsymbol{sobran}}(\boldsymbol{T}_i,\boldsymbol{C}_j)}{\boldsymbol{Precio}\,(\boldsymbol{T}_i)} \quad \textbf{(Ecuación 15)}$$

Donde

T_i ≡ cada modelo de FPGA (1 ≤ i ≤ m). Para este trabajo m=246.
C_j ≡ cada IP del PdC (1 ≤ j ≤ n). Para este trabajo n=16.

	Circuitos			
	C_1	C_2	…	C_n
T_1	$P4_{1,1}$	$P4_{1,2}$	…	$P4_{1,n}$
T_2	$P4_{2,1}$	$P4_{2,2}$	…	$P4_{2,n}$
…	…	…	…	…
T_m	$P4_{m,1}$	$P4_{m,1}$	…	$P4_{m,n}$

Tabla 53.Puntuación P4 que relaciona %bits de bloques multiplicadores sobrantes con precio por unidad

Estimación de consumo versus precio (P5)

El último parámetro compara en términos relativos el consumo estimado por los analizadores de potencia. La puntuación P5 relaciona el consumo estimado con el precio en cada modelo de FPGA. El mejor resultado será aquel que al menor precio consiga un menor consumo.

$$\mathbf{P5}_{i,j} = \frac{1/Potencia_{estimada}(T_i,C_j)}{Precio\,(T_i)} \quad \textbf{(Ecuación 16)}$$

Donde

T_i ≡ cada modelo de FPGA (1 ≤ i ≤ m). Para este trabajo m=246.
C_j ≡ cada IP del PdC (1 ≤ j ≤ n). Para este trabajo n=16.

	Circuitos			
	C_1	C_2	…	C_n
T_1	$P5_{1,1}$	$P5_{1,2}$	…	$P5_{1,n}$
T_2	$P5_{2,1}$	$P5_{2,2}$	…	$P5_{2,n}$
…	…	…	…	…
T_m	$P5_{m,1}$	$P5_{m,1}$	…	$P5_{m,n}$

Tabla 54.Puntuación P5 que relaciona la potencia estimada con el precio por unidad

El resto de magnitudes como el tiempo de establecimiento o mantenimiento externo no participarán en esta primera propuesta, son parámetros relacionados con el interconexionado con otros chips y en esta ocasión la puntuación se limitará a valores internos de la FPGA.

En cuanto a incluir en la puntuación el tiempo de CPU o la memoria de pico empleada, se sopesó la posibilidad de incorporarla como un factor de corrección a la puntuación final. Finalmente, se desechó la idea al tratarse de una valoración relacionada con el equipo de trabajo y no con la FPGA en sí.

Aunque estas magnitudes no forman parte del sistema de puntuaciones finalmente establecido, se dispone de ellas en la base de datos que almacena la información leída tras la ejecución del PdC.

5.2.2. Normalización de las puntuaciones (Pk')

Para que todas las puntuaciones, en adelante Pk, ponderen en igual proporción sobre la global y la final, se establece una correspondencia entre cada rango calculado por cada puntuación y una valoración de 1 a 10. En general, todas las Pk evolucionan en todos los circuitos siguiendo una tendencia lineal con coeficiente de determinación múltiple (r^2) en torno a 0,83, ver ejemplo en Figura 52. Es consistente realizar la conversión de las Pk a Pk′, nombre de cada Pk normalizado, a partir de la ecuación de la recta que se determine por cada Pk y circuito.

Es justo decir que los modelos de FPGAs más económicos, unos 3 o 4 dispositivos, se ven perjudicados por la conversión lineal. Son ellos los que alejan a r^2 de un valor más cercano a 1 y esto se debe a su reducido precio y buenas prestaciones. Sin embargo,

aunque esta conversión minore su distancia en puntuación respecto a otros modelos, seguirán manteniendo una posición adelantada, y es suficiente para establecer una clasificación.

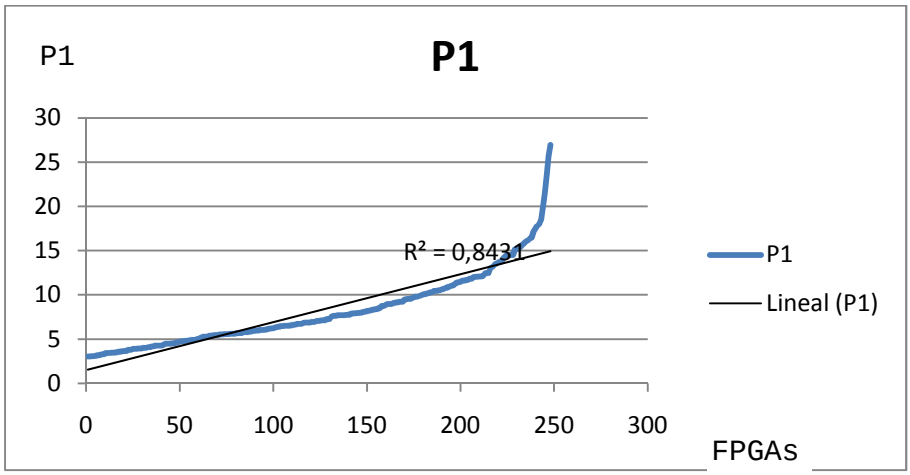


Figura 52. Evolución de P1 correspondiente al circuito SC_CORPRO (azul), línea de tendencia (negro) y coeficiente de determinación múltiple (R^2)

El procedimiento a seguir para conseguir la normalización de las Pk por cada uno de los 16 circuitos del PdC y cada una de las cinco puntuaciones consiste en:

- Determinar el valor máximo y mínimo de cada Pk
- Calcular la ecuación de la recta que contiene a los puntos (Pk_{min},1) y (Pk_{max},10). Figura 53
- Calcular la correspondencia Pk′ de todos los valores de Pk.

- Repetir para el resto de puntuaciones (Pk) y circuitos (Ci)

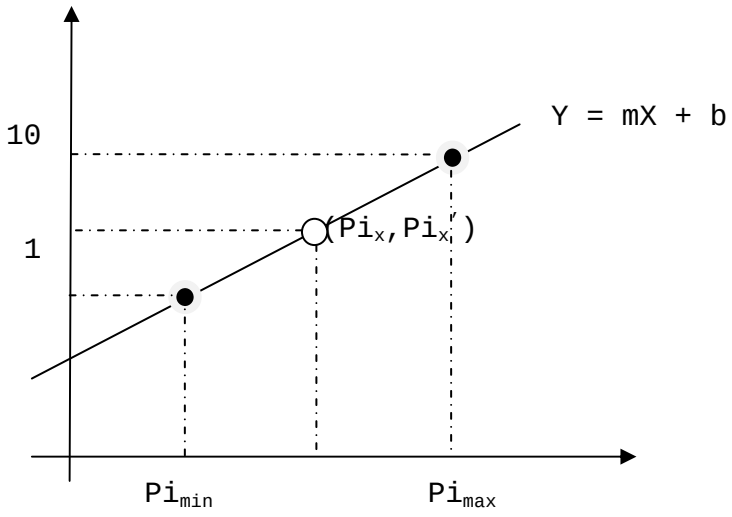


Figura 53. Ecuación de la recta que relaciona Pi con Pi´

Al término de este proceso, cada FPGA tendrá una puntuación de 1 a 10 por circuito y parámetro, otorgándose un máximo de 10 a la que tenga mejor relación parámetro/precio.

En este momento, ya es posible establecer varias clasificaciones ordenadas de la muestra de FPGAs dependientes de cada una de las cinco puntuaciones.

5.2.3. Cálculo de la puntuación global (PG)

La puntuación global ofrece información sobre el comportamiento general de cada FPGA ante un determinado circuito del PdC aunando todas las puntuaciones obtenidas individualmente para ese circuito y tecnología.

Esta puntuación se calcula como la media geométrica, ecuación 17, de las Pk’ de cada FPGA ante un circuito del PdC, de este modo queda premiada la regularidad de todas ante el despunte de alguna de ellas, ofreciendo una puntuación más equilibrada que si se utilizara la media aritmética.

$$\boldsymbol{PG_{i,j}} = \sqrt[5]{\prod_{k=1}^{5} \boldsymbol{Pk'_{i,j}}} \quad \textbf{(Ecuación 17)}$$

5.2.4. Cálculo de la puntuación final (PF)

El objetivo final del sistema de puntuaciones es clasificar las FPGAs de la muestra según su comportamiento ante los circuitos del PdC.

En el paso anterior se obtuvo una puntuación global por cada FPGA y circuito, resta por tanto relacionarlas todas para llegar a una única puntuación final que resuma el éxito de cada modelo frente al PdC.

Es importante tener en cuenta que no todos los circuitos del PdC se implementaron con éxito en todos los dispositivos objetivo. En algunos casos las FPGA no contaban con recursos suficientes para ello y en otras hubo otros fallos durante la ejecución automática.

Para no distorsionar los resultados, el cálculo de la puntuación final se hará según tres posibles casos:

Caso 1:

Considerar solo aquellas FPGAs de la muestra sobre las que se ha conseguido implementar todos los circuitos del PdC

Caso 2:

Considerar solo aquellos circuitos que se implementaron sobre todas las FPGAs.

Caso 3:

Seleccionar los circuitos de interés y puntuar solo las tecnologías sobre las que se implementaron todos. Este será el caso empleado si se desea obtener una puntuación por sector comercial.

Una vez determinada la muestra a la que se le calculará la puntuación final, caso 1, caso 2 ó caso 3, se vuelve a utilizar la media geométrica pero utilizando en esta ocasión las puntuaciones globales PG oportunas.

Para el caso 1, PF[25] se calcula como la media geométrica de los PG obtenidos anteriormente para cada FPGA que haya implementado todos los IPs del PdC.

$$\boldsymbol{PF_i = \sqrt[n]{\prod_{j=1}^{n} PG_{i,j}}} \quad \textbf{(Ecuación 18)}$$

Donde se cumple $i \leq m = 246$

[25] La letra i es el índice asignado a las tecnologías de la muestra, para este trabajo toma un valor de 246. El índice j hace referencia a los circuitos del PdC, sin agrupar en sectores suma un total de 16.

Para el caso 2, se calcula la media geométrica de las puntuaciones globales de los IPs implementados en todas las FPGAs.

$$\boldsymbol{PF_i} = \sqrt[p]{\prod_{j=1}^{p} \boldsymbol{PG_{i,j}}} \quad \textbf{(Ecuación 19)}$$

$$p \leq n = 16$$

Este caso permite clasificar todas las FPGAs de la muestra aunque usando como referencia un número de circuitos inferior al del PdC completo.

El caso 3 es una combinación de los dos anteriores y su cálculo responde a la siguiente ecuación:

$$\boldsymbol{PF_i} = \sqrt[q]{\prod_{j=1}^{q} \boldsymbol{PG_{i,j}}} \quad \textbf{(Ecuación 20)}$$

$$i \leq m = 246\,; q \leq n = 216$$

PASO	Nombre de la puntuación	Identificador	Cálculo
1	Puntuación por parámetro	$Pk_{i,j}$ (k ∈[1,5])	P1 = F (MHZ)/Precio($) P2 = %U.L.sobran/Precio($) P3 = %Bits_sobran/Precio($) P4 = %Mult_sobran/ Precio($) P5 = (1/Potencia_estimada(mW)) /Precio($)
2	Normalización (Conversión lineal) (1 a 10)	$Pk'_{i,j}$ (k ∈[1,5])	$m = {}^{9}/_{(Pk_{max} - Pk_{min})}$ $b = {}^{(Pk_{max} - 10 * Pk_{min})}/_{(Pk_{max} - Pk_{min})}$ $Pk' = m * Pk + b$

PASO	Nombre de la puntuación	Identificador	Cálculo	
3	Puntuación global (Media Geométrica)	$PG_{i,j}$	$PG_{i,j} = \sqrt[5]{\prod_{k=1}^{5} Pk'_{i,j}}$	
4	Puntuación final (Media Geométrica)	PF_i	Todos los circuitos i<m	$PF_i = \sqrt[n]{\prod_{j=1}^{n} PG_{i,j}}$
			Todas las FPGAs j < n	$PF_i = \sqrt[p]{\prod_{j=1}^{p} PG_{i,j}}$
			Selección de circuitos i<m j<n	$PF_i = \sqrt[q]{\prod_{j=1}^{q} PG_{i,j}}$

Tabla 55. Resumen del sistema de puntuaciones

6. Desarrollo de Aplicaciones

6.1. Objetivos de las aplicaciones

Como ya se comentó en el capítulo de introducción, la ejecución del PdC supone manejar una gran cantidad de datos, ya que 16 circuitos serán implementados sobre un total de 186 modelos de FPGAs. Al término de todas las operaciones será necesario abrir una media de 4 ficheros por cada combinación circuito-FPGA, 11.904 en total, y leer ciertos valores en ellos. Sólo el número de ficheros a abrir ya argumenta la automatización de esta tarea. Además la información extraída debe ser tratada, organizada y almacenada dado que más tarde se hará uso de ella para el cálculo de puntuaciones y la clasificación de FPGAs.

La primera de las aplicaciones se denomina **"Resultados_PdC"** y se desarrolló con el objetivo de leer todos los valores significativos de los múltiples informes generados por las herramientas de Xilinx y Altera. Homogeneiza todos los parámetros para que muestren la información con el mismo formato y unidades, efectúa los cálculos necesarios hasta conseguir las magnitudes objetivo a partir de las cuales calcular las cinco puntuaciones Pk y por último, almacena todo en una base de datos.

La segunda recibe el nombre **"Asistente_FPGA"** y presenta una serie de criterios, coherentes con la muestra de FPGAs analizadas, que filtran el conjunto total según características constructivas y precio anexándoles sus correspondientes puntuaciones.

La herramienta de desarrollo utilizada ha sido Microsoft Visual Studio que es un entorno integrado para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic.NET. A partir de la versión 2005 Microsoft ofrece gratuitamente las Express Editions, iguales al entorno comercial pero sin las características más avanzadas.

Las aplicaciones informáticas para esta tesis se han realizado con Visual Basic Express Edition 2008, con potencia suficiente para realizar adecuadamente todas las tareas necesarias. Para la creación y gestión de la base de datos se utilizó Microsoft Access con licencia campus de la Universidad de Cádiz.

6.2. Aplicación 1: Lectura de resultados del PdC

Esta aplicación realiza tres acciones principales, en la primera, avanza por la jerarquía de carpetas de resultados localizando en cada fichero los parámetros determinados en el capítulo anterior. Toda lo información leída es almacenada en una tabla de la base de datos, en adelante BD, y si alguno de los informes no es localizado, se advierte de ello en otra tabla. Dado el caso de que la etapa de emplazamiento sea incapaz de ajustar el circuito a un determinado modelo de FPGA lo notifica en una tercera tabla.

En segundo lugar, el programa realiza el cálculo de las puntuaciones Pk y a continuación las normaliza para mantener sus valores en el rango de 1 a 10, Pk’. Si detecta algún fallo durante este proceso lo consignará en la BD.

Por último, ofrece la posibilidad de ver qué circuitos se ajustaron en todas las FPGAs y permite añadir los que corresponderán a cada uno de los tres sectores comerciales, pasando todo después a ser almacenados en la BD.

En los siguientes apartados se presentan las ideas clave barajadas durante el desarrollo de la aplicación, para aclarar la utilización del mismo se adjunta un pequeño manual de ayuda en el anexo D.

6.2.1. Primera tarea: Lectura de ficheros

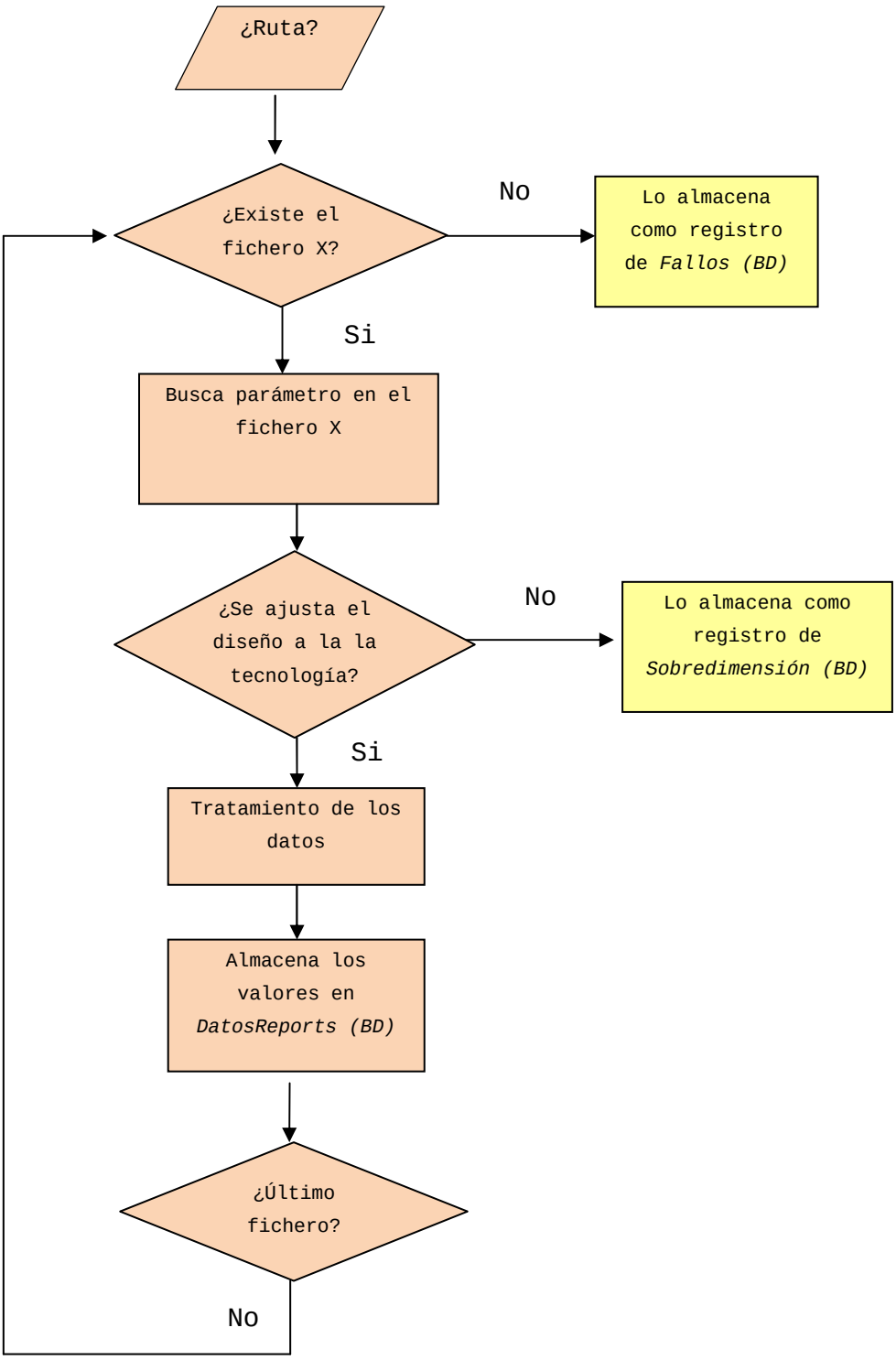


Figura 54. Diagrama que describe el algoritmo de la lectura de ficheros y datos

El proceso se inicia especificando la ruta donde se almacenaron los resultados del PdC, y continúa tal y como se muestra en el diagrama de la Figura 54. Las tablas que componen la BD se encuentran detalladas en el anexo C.

6.2.1.1 Informes y valores

Todos los datos necesarios para calcular las puntuaciones Pk se encuentran en los informes y se localizan mediante la búsqueda de una determinada cadena de caracteres, diferente según parámetro y fabricante.

Se han creado varias funciones con el objetivo de realizar la búsqueda de datos, cada una recibe como argumento la cadena de texto adecuada y devuelve el valor correspondiente.

A continuación se cita el nombre de los ficheros que hay que abrir y la cadena de texto que se usa como argumento, marcada en negrita, para las funciones de búsqueda de datos.

Informes de ISE 10.1 (Webpack)

a) Informe de la síntesis lógica (SRP)

Es el documento generado tras la fase de síntesis, y contiene una primera valoración de los recursos a utilizar en la FPGA en base a las primitivas instanciadas por el sintetizador lógico. Para este trabajo esta información no es importante y de él se extrae solo el tiempo empleado en esta tarea así como el pico de memoria necesario. Aparecen en el fichero tal y como se detalla en el siguiente ejemplo:

***Total CPU time to Xst completion**: 44.18 secs*
***Total memory usage is** 206812 kilobytes*

b) Informe del proceso de mapeado (MAP)

Solo interesa la información de tiempo de CPU y memoria empleada en esta tarea:

***Peak Memory Usage**: 164 MB*
***Total CPU time to MAP completion**: 3 secs*

c) Informe del emplazamiento e interconexionado (PAR)

Generado tras el emplazamiento y ruteado incluye toda la información sobre los recursos finalmente empleados en la implementación del diseño. De él se extrae las E/S, los bloques de memoria, los slices o unidades lógicas y los multiplicadores, para todos aparece el valor de recursos utilizados y disponibles. El siguiente ejemplo aclara su formato:

Device Utilization Summary:

Number of BUFGMUXs	1 out of 8	12%
Number of External IOBs	31 out of 124	25%
Number of LOCed IOBs	0 out of 31	0%
Number of RAMB16s	1 out of 4	25%
Number of Slices	915 out of 768	119%
Number of SLICEMs	169 out of 384	44%
Number of MULT18X18s[26]	2 out of 4	50%

La información de tiempo y memoria de esta fase debe buscarse en este mismo documento:

Total CPU time to PAR completion: 1 mins 51 secs
Peak Memory Usage: 288 MB

[26] Según el modelo de FPGA el texto a localizar en la búsqueda de multiplicadores puede variar según indica la Tabla 49.

d) **Informe del análisis estático de tiempo (TWR)**

Concluida la etapa de implementación, los ficheros por lotes lanzan el analizador de tiempo estático para evaluar el camino crítico del diseño, así como los requisitos temporales de las entradas al circuito o el retardo de las salidas del mismo. Todos estos valores pueden extraerse del fichero TWR localizando las siguientes líneas (en negrita):

```
Setup/Hold to clock CLK_I
------------+------------+------------+------------------+--------+
            |  Setup to  |  Hold to   |                  | Clock  |
Source      | clk (edge) | clk (edge) |  Internal Clock(s) | Phase  |
------------+------------+------------+------------------+--------+
CE_I        |   4.418(R)|   1.157(R)|CLK_I_BUFGP       |  0.000|
DATA_I<0>   |  -0.422(R)|   1.332(R)|CLK_I_BUFGP       |  0.000|
DATA_I<1>   |  -0.480(R)|   1.378(R)|CLK_I_BUFGP       |  0.000|
DATA_I<2>   |  -0.422(R)|   1.332(R)|CLK_I_BUFGP       |  0.000|
DATA_I<3>   |  -0.480(R)|   1.378(R)|CLK_I_BUFGP       |  0.000|
DATA_I<4>   |  -0.422(R)|   1.332(R)|CLK_I_BUFGP       |  0.000|
DATA_I<5>   |  -0.480(R)|   1.378(R)|CLK_I_BUFGP       |  0.000|
DATA_I<6>   |  -0.422(R)|   1.332(R)|CLK_I_BUFGP       |  0.000|
DATA_I<7>   |  -0.480(R)|   1.378(R)|CLK_I_BUFGP       |  0.000|
KEY_I<0>    |  -0.422(R)|   1.332(R)|CLK_I_BUFGP       |  0.000|
KEY_I<1>    |  -0.480(R)|   1.378(R)|CLK_I_BUFGP       |  0.000|
KEY_I<2>    |  -0.422(R)|   1.332(R)|CLK_I_BUFGP       |  0.000|
KEY_I<3>    |  -0.480(R)|   1.378(R)|CLK_I_BUFGP       |  0.000|
KEY_I<4>    |  -0.422(R)|   1.332(R)|CLK_I_BUFGP       |  0.000|
KEY_I<5>    |  -0.480(R)|   1.378(R)|CLK_I_BUFGP       |  0.000|
KEY_I<6>    |  -0.422(R)|   1.332(R)|CLK_I_BUFGP       |  0.000|
KEY_I<7>    |  -0.480(R)|   1.378(R)|CLK_I_BUFGP       |  0.000|
RESET_I     |   3.813(R)|   1.157(R)|CLK_I_BUFGP       |  0.000|
VALID_DATA_I|   4.680(R)|   1.332(R)|CLK_I_BUFGP       |  0.000|
VALID_KEY_I |   1.419(R)|   1.128(R)|CLK_I_BUFGP       |  0.000|
------------+------------+------------+------------------+--------+
Clock CLK_I to Pad
------------+------------+------------------+--------+
            | clk (edge) |                  | Clock  |
Destination |   to PAD   |Internal Clock(s) | Phase  |
------------+------------+------------------+--------+
DATA_O<0>   |   9.894(R)|CLK_I_BUFGP       |  0.000|
DATA_O<1>   |   9.523(R)|CLK_I_BUFGP       |  0.000|
DATA_O<2>   |  10.048(R)|CLK_I_BUFGP       |  0.000|
DATA_O<3>   |   9.613(R)|CLK_I_BUFGP       |  0.000|
DATA_O<4>   |   9.894(R)|CLK_I_BUFGP       |  0.000|
DATA_O<5>   |   9.961(R)|CLK_I_BUFGP       |  0.000|
DATA_O<6>   |   9.894(R)|CLK_I_BUFGP       |  0.000|
DATA_O<7>   |   9.967(R)|CLK_I_BUFGP       |  0.000|
KEY_READY_O |   7.809(R)|CLK_I_BUFGP       |  0.000|
VALID_O     |   9.961(R)|CLK_I_BUFGP       |  0.000|
------------+------------+------------------+--------+

Clock to Setup on destination clock CLK_I
---------------+----------------+---------+---------+---------+
               | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock   |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
---------------+----------+---------+---------+---------+
CLK_I          |  10.866|        |        |        |
---------------+----------+---------+---------+---------+
```

La función empleada en este caso, no solo extrae la información de las tablas, sino que tambien ordena su contenido para determinar el caso más desfavorable, ya que este no es presentado directamente por ISE.

Este informe no especifica el tiempo de CPU empleado aunque si la memoria de pico necesaria:

Peak Memory Usage: *108 MB*

e) Informe del análisis de potencia (PWR)

Por último la estimación de consumo, en milivatios (mW), del diseño puede consultarse en el fichero tipo PWR. Los datos de interés son buscados localizando las cadenas de texto siguiente:

Total estimated power consumption *| | 30 |*

Informes de Quartus II 9 (Web edition)

a) Informe del análisis estático de tiempo (TAO)

Devuelve los tiempos de establecimiento, mantenimiento y de propagación, asi como uno de los sumandos necesarios para obtener el tiempo máximo entre flip-flops. Aunque el informe incluye todos los caminos analizados por defecto, los ordena comenzando por el más desfavorable para cada parámetro. La función de búsqueda puede localizar directamente el dato de interés sin más que leer el primer valor.

```
------------------------------------------------------------------------------
Clock Setup: 'CLK_I'
------------------------------------------------------------------------------
Path Number          : 1
[...]
Actual Longest P2P Time     : 10.934 ns
[...]
------------------------------------------------------------------------------
tsu
------------------------------------------------------------------------------
```

Path Number : 1
[…]
***Actual tsu** : 8.708 ns*
[…]
----
tco

Path Number : 1
[…]
***Actual tco** : 6.932 ns*
[…]
----
th

Path Number : 1
[…]
***Actual th** : 0.180 ns*
[…]

b) Informe de análisis estático de tiempo (TAN.RPT)

Para completar el cálculo del máximo retardo entre dos flip-flops, se necesita extraer información adicional de este fichero. En concreto, en él se buscarán los tiempos de establecimiento y de propagación internos de los flip-flops de cada tecnología. Estos valores sumados al "Actual longest P2P time" obtenido anteriormente darán como resultado el tiempo de retardo máximo entre dos flips-flops por cada diseño. El resultado de la suma será almacenado en la BD.

Estos valores aparecen en el apartado denominado *Timing Analyzer Messages* del fichero de texto:

+----------+
; Timing Analyzer Messages ;
+----------+
[…]
***Info: + Micro clock to output delay of source is** 0.209 ns*
***Info: + Micro setup delay of destination is** -0.036 ns*
[…]

c) Informe del análisis de potencia (POW.SUMMARY)

Una vez abierto puede conocerse la estimación de potencia localizando el siguiente texto:

***Total Thermal Power Dissipation** : 38.26 mW*

d) Informe del flujo de trabajo (FLOW.RPT)

A diferencia de la herramienta de Xilinx, Altera recopila en un fichero toda la información relativa a los recursos utilizados, los tiempos de CPU, y los picos de memoria de la síntesis e implementación.

Para determinar los recursos lógicos, pines, memoria, unidades lógicas y multiplicadores, se utilizará una función a la que se le pasará el texto que actuará como argumento, se destacan en negrita en el ejemplo siguiente:

Total logic elements *: 2,254 / 4,608 (49 %)*
Total combinational functions : 2,124 / 4,608 (46 %)
Dedicated logic registers : 487 / 4,608 (11 %)
Total registers : 487
Total pins *: 31 / 158 (20 %)*
Total virtual pins : 0
Total memory bits *: 4,096 / 119,808 (3 %)*
Embedded Multiplier 9-bit elements *: 0 / 26 (0 %)*[27]

```
+-----------------------------------------------------------------------------------------------------------------------------+
; Flow Elapsed Time                                                                                                           ;
+-------------------------+---------------+-------------------------+------------+-----------------------------------+
; Module Name             ; Elapsed Time ;  Average;                ; Peak Virtual;  Total CPU
                                            Processors               memory          time
                                            Used;
+-------------------------+---------------+-----------------+-------------------+-----------------------------------+
; Analysis & Synthesis    ; 00:00:47     ; 1.0             ; 208 MB            ; 00:00:37
; Fitter                  ; 00:00:57     ; 1.0             ; 251 MB            ; 00:00:48
; Assembler               ; 00:00:14     ; 1.0             ; 193 MB            ; 00:00:03
; Classic Timing Analyzer ; 00:00:16     ; 1.0             ; 200 MB            ; 00:00:06
; PowerPlay Power Analyzer ; 00:00:14    ; 1.0             ; 210 MB            ; 00:00:04
; Total                   ; 00:02:28     ; --              ; --                ;00:01:38
```

6.2.1.2 Tratamiento de los datos

En algunas de las magnitudes leídas se da el caso de que cada fabricante presenta los valores usando formato y unidades diferentes. A continuación se relacionan mediante una tabla la conversión practicada por la aplicación para conseguir un único modelo de presentación en cada conjunto de datos. La última

[27] A la función de búsqueda se le pasará también la cadena ***"Embedded Multiplier 18-bit elements"*** para el caso de que el bloque multiplicador se haya utilizado con entradas de 18 bits.

columna indica las unidades y formato final a que se convierte cada parámetro de forma previa a su almacenamiento en la tabla *DatosReports* de la base de datos.

	Parámetro	Xilinx	Altera	Formato final
Síntesis	Tiempo de CPU	Segundos	hh:mm:ss	Segundos
	Memoria de pico	Kilobytes	Megabytes	Megabytes (MB)
Implementación	Tiempo de CPU	Minutos y Segundos	hh:mm:ss	Segundos
	Memoria de pico	Megabytes	Megabytes	Megabytes
	Bloques de memoria	Número de bloques utilizados	Bits	Bits
	Multiplicadores	Número de bloques de 18x18 bits	Número de bloques de 18x18 bits ó de 9x9 bits	Número de bloques de 18x18 bits

Tabla 56. Formatos y unidades finales

Cada registro de la citada tabla incluirá los campos fabricante, tecnología, estrategia, sector, unidades lógicas usadas y disponibles, bits de bloque usados y disponibles, multiplicadores de 18x18 bits usados y disponibles, pines usados y disponibles, tiempo de retardo del camino crítico entre registros, tiempo de establecimiento, tiempo de mantenimiento, retardo de propagación, tiempo de síntesis, tiempo de implementación y pico de memoria.

6.2.1.3 Detección de fallos de ejecución del PdC

Durante esta primera fase, la aplicación detecta dos tipos de anomalías, la ausencia del fichero de información buscado, y la imposibilidad de concluir la fase de implementación con éxito.

a) Inexistencia de algún fichero

Antes de analizar cada informe, se comprueba la existencia del mismo, ya que si no se han seguido los pasos recomendados a la hora de preparar el PdC, la ejecución puede fallar en algunas de sus fases.

Otra de las causas posibles de ausencia de algún fichero es el inadecuado tamaño de una FPGA para alojar un diseño grande, por escasez de unidades lógicas o bien de pines de entrada/salida. En este caso, tendrá lugar la fase de síntesis o incluso la de implementación, pero no el análisis de tiempo ni el de potencia.

En la tabla *Fallos* de la BD, se almacenará fabricante, circuito, estrategia, tecnología y la ruta completa del fichero fallido.

b) Fallo de implementación

Se refiere a aquellos diseños que son demasiado grandes para implementarse sobre las FPGAs más pequeñas, esto es, con pocas unidades lógicas o escasas entradas/salidas. El método empleado consiste en, leer la cantidad de unidades lógicas y entradas/salidas empleadas en los informes de implementación y comprobar si alguna de ellas o ambas son mayores que la cantidad de unidades lógicas y entradas/salidas disponibles.

Si la comprobación resulta positiva, se añade a la tabla S*obredimensionados* de la BD. El registro incluirá los campos circuitos, fabricante, tecnología, unidades lógicas disponibles y entradas/salidas tanto disponibles como utilizadas.

La Figura 55 presenta el aspecto de la pantalla que muestra las tablas con los datos leídos, los fallos de fichero y los no implementados por falta de recursos.

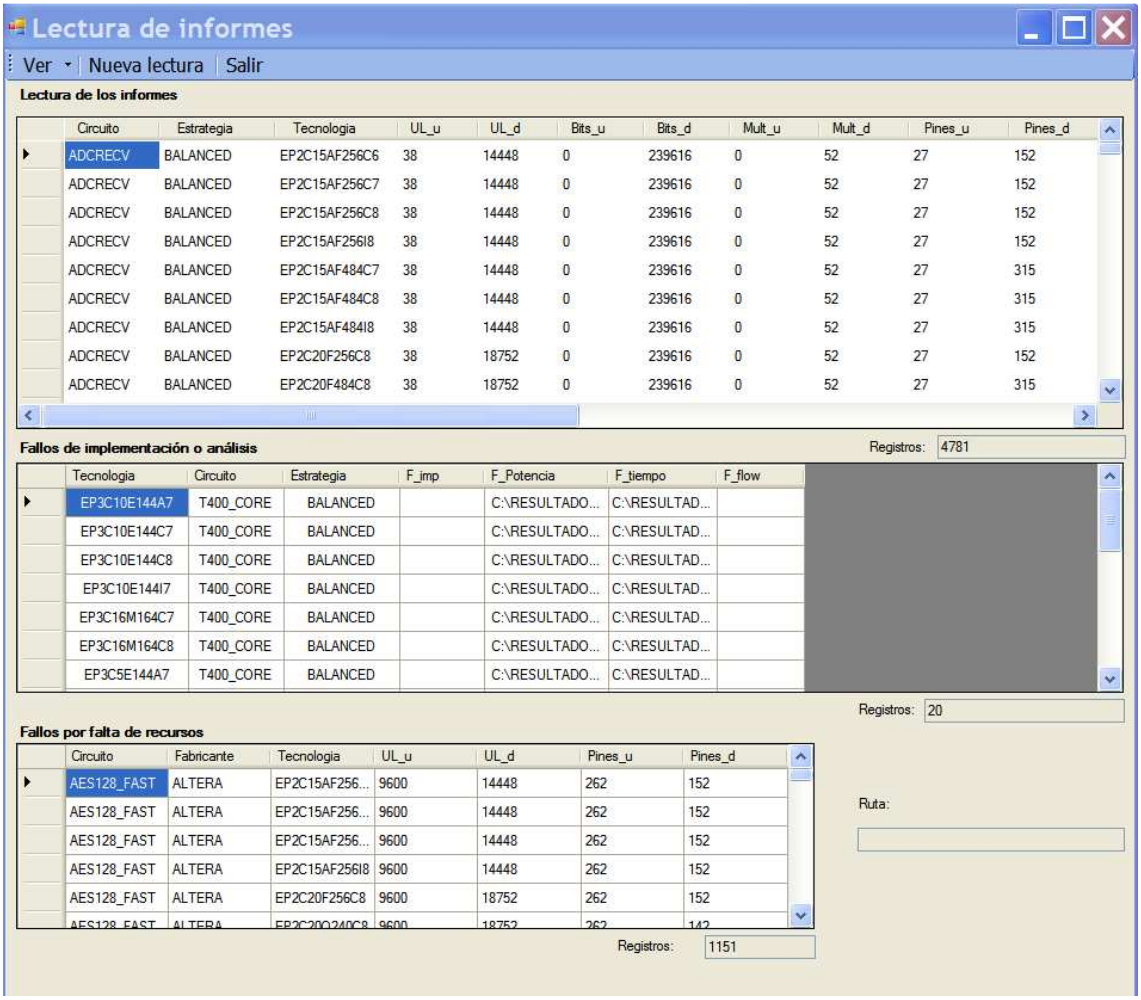


Figura 55. Pantalla de presentación de la lectura de datos y de los fallos encontrados

Mediante el menú que figura en esta pantalla se tiene acceso a las restantes ventanas que gestiona el programa, al inicio de una nueva lectura de datos o bien a salir de la aplicación.

6.2.2. Segunda tarea: Circuitos del PdC

Para generalizar la utilización de esta herramienta, se añade a la aplicación la capacidad de determinar qué circuitos integran el PdC. De este modo, si el conjunto de núcleos del PdC se ve modificado o incrementado, el programa continuará realizando su trabajo adecuadamente. El nombre de los circuitos queda almacenado en la tabla *Circuitos* de la BD, Tabla 57.

Nombre_circuito
ADCRECV
AES128_FAST
AES_DEC
AES_ENC
SIN
SPI_BOOT
T400_CORE
ANALYTIC_FILTER_H_A1
DDS_SYNTHESIZER
FPU
I2CSLAVETOP
RS_DEC
RS_ENCODE
UART_16750
UCRC_PAR
SC_CORPROC

Tabla 57. Tabla Circuito de la BD incluyendo los núcleos del PdC de este trabajo

6.2.3. Tercera tarea: Cálculo de las puntuaciones

6.2.3.1 Puntuaciones Pk

Una vez que se dispone de toda la información se procede al cálculo de puntuaciones previo cómputo de las magnitudes necesarias, Tabla 58:

Magnitud	Cálculo	Campos de DatosReports (BD)	Pk
Frecuencia máxima (MHz)	$\frac{1}{\text{Tiempo máximo entre FF (µs)}}$	$\frac{1}{Reg2Reg}$	P1=F_{max}/$
Unidades lógicas restantes (%)	$\frac{U.L._{disponibles} - U.L._{utilizadas}}{U.L._{disponibles}} \times 100$	$\frac{UL\_d - UL\_u}{UL\_d} \times 100$	P2=U.L.$_{sobran}$/$
Bits memoria restantes (%)	$\frac{Bits_{disponibles} - Bits_{utilizadas}}{Bits_{disponibles}} \times 100$	$\frac{Bits\_d - Bits\_u}{Bits\_d} \times 100$	P3=bits$_{sobran}$/$
Bloques multiplicadores restantes (%)	$\frac{Mult_{disponibles} - Mult_{utilizadas}}{Mult_{disponibles}} \times 100$	$\frac{Mult\_d - Mult\_u}{Mults\_d} \times 100$	P4=Mult$_{sobran}$/$

Tabla 58. Cálculos que realiza la aplicación antes de obtener Pk

La quinta magnitud es la potencia, para su presentación en la tabla correspondiente no necesita ninguna conversión y aparece utilizando la unidad mW. Sin embargo, para calcular P5 se hará uso de la potencia expresada en W, evitándose así el manejo de cifras demasiado pequeñas y por ende posibles errores de redondeo.

Todas las puntuaciones obtenidas quedan almacenadas en la tabla *Puntuaciones* de la BD que incluye los campos circuito, estrategia, tecnología, frecuencia,unidades lógicas restantes,bits restantes, multiplicadores restantes, potencia, precio, tiempo de CPU, memoria de pico,P1,P2,P3,P4,P5.

6.2.3.2 Puntuación normalizada Pk'

La puntuación debe normalizarse en el rango 1 a 10, para ello, tal y como ya se expuso, se opta por utilizar una ecuación de regresión lineal, calculando para cada puntuación Pk de cada uno de los 16 circuitos del PdC los coeficientes m y b.

Para el cálculo de m y b, se ha desarrollado una función que obtiene estos valores tal y como se indica en la Tabla 59.

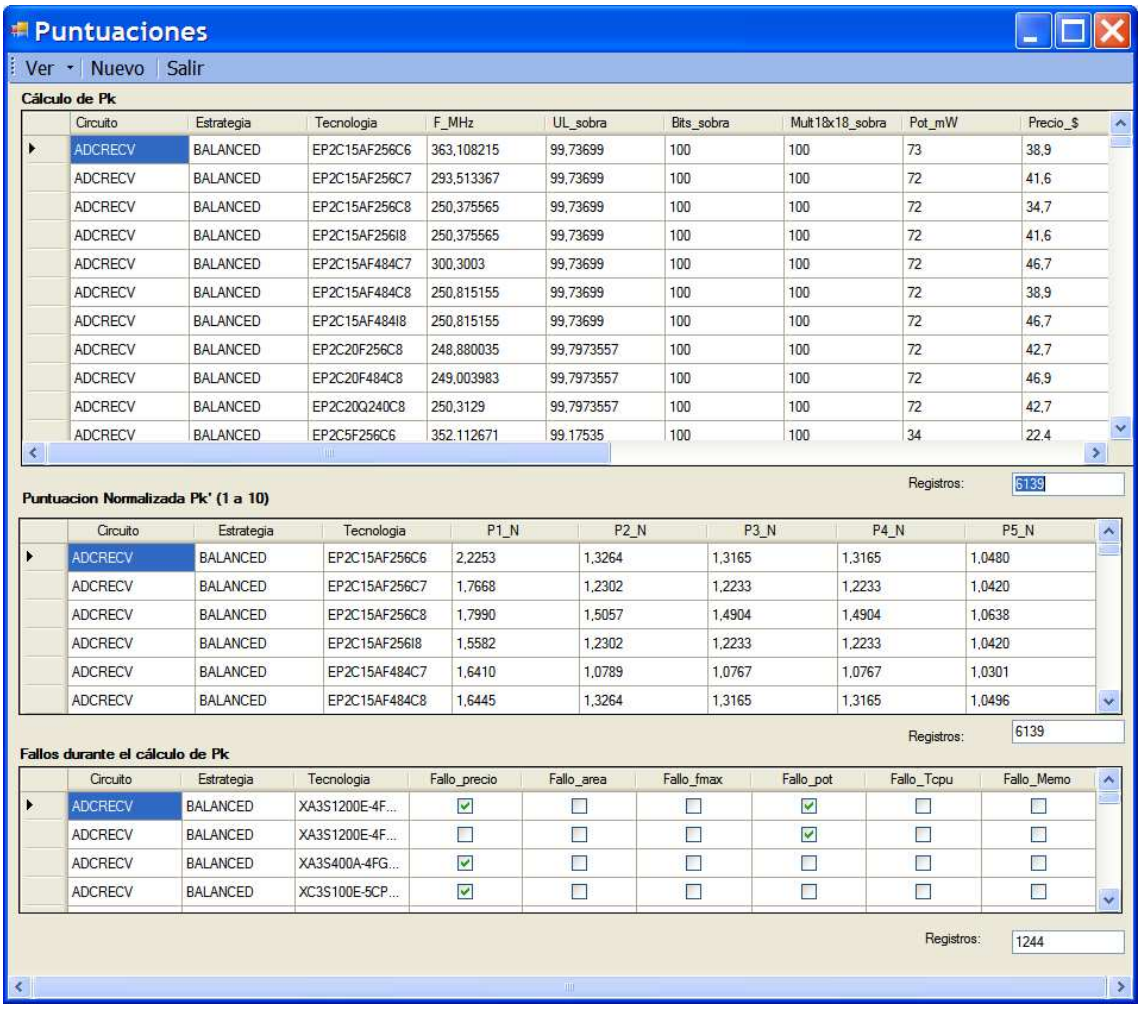
Variable calculada	Fórmula de cálculo	Donde
Pk_{max} y Pk_{min}	Se crea una función que recorre la tabla *Puntuaciones* y determina el valor máximo y minimo por circuito y puntuación Pk.	
m	$\frac{9}{Pk_{max}(Ci) - Pk_{\min}(Ci)}$	i = 1,2…16 K = 1,2..5
b	$\frac{Pk_{max}(Ci) - 10 * Pk_{min}(Ci)}{Pk_{max}(Ci) - Pk_{min}(Ci)}$	i = 1,2…16 K = 1,2..5
Pk’	Pk’=m*Pk + b	K = 1,2..5

Tabla 59. Relación de fórmulas previas al cálculo de Pk’

6.2.3.3 Fallos durante el cálculo de las puntuaciones

Igual que en la etapa anterior, durante el cálculo de puntuaciones es posible que se produzca algún error. Estos serán detectados y almacenados en la tabla *Fallo_puntuaciones* de la BD.

Los errores pueden presentarse en el caso de que no se haya especificado el precio de algunas de las tecnologías de la muestra, o bien porque durante el cálculo de las magnitudes que determinan Pk se detecte que el valor leído no es consistente. Por ejemplo, durante el cálculo de la frecuencia máxima se generará un error si el tiempo entre flip-flops se almacenó con valor 0. La Figura 56 muestra la vista de las tablas *Puntuaciones, Normalizadas y Fallos_puntuaciones* que ofrece la aplicación al término de esta etapa.



Puntuaciones

Ver | Nuevo | Salir

Cálculo de Pk

Circuito	Estrategia	Tecnologia	F_MHz	UL_sobra	Bits_sobra	Mult18x18_sobra	Pot_mW	Precio_$
ADCRECV	BALANCED	EP2C15AF256C6	363,108215	99,73699	100	100	73	38,9
ADCRECV	BALANCED	EP2C15AF256C7	293,513367	99,73699	100	100	72	41,6
ADCRECV	BALANCED	EP2C15AF256C8	250,375565	99,73699	100	100	72	34,7
ADCRECV	BALANCED	EP2C15AF256I8	250,375565	99,73699	100	100	72	41,6
ADCRECV	BALANCED	EP2C15AF484C7	300,3003	99,73699	100	100	72	46,7
ADCRECV	BALANCED	EP2C15AF484C8	250,815155	99,73699	100	100	72	38,9
ADCRECV	BALANCED	EP2C15AF484I8	250,815155	99,73699	100	100	72	46,7
ADCRECV	BALANCED	EP2C20F256C8	248,880035	99,7973557	100	100	72	42,7
ADCRECV	BALANCED	EP2C20F484C8	249,003983	99,7973557	100	100	72	46,9
ADCRECV	BALANCED	EP2C20Q240C8	250,3129	99,7973557	100	100	72	42,7
ADCRECV	BALANCED	EP2C5F256C6	352,112671	99,17535	100	100	34	22,4

Registros: 6139

Puntuacion Normalizada Pk' (1 a 10)

Circuito	Estrategia	Tecnologia	P1_N	P2_N	P3_N	P4_N	P5_N
ADCRECV	BALANCED	EP2C15AF256C6	2,2253	1,3264	1,3165	1,3165	1,0480
ADCRECV	BALANCED	EP2C15AF256C7	1,7668	1,2302	1,2233	1,2233	1,0420
ADCRECV	BALANCED	EP2C15AF256C8	1,7990	1,5057	1,4904	1,4904	1,0638
ADCRECV	BALANCED	EP2C15AF256I8	1,5582	1,2302	1,2233	1,2233	1,0420
ADCRECV	BALANCED	EP2C15AF484C7	1,6410	1,0789	1,0767	1,0767	1,0301
ADCRECV	BALANCED	EP2C15AF484C8	1,6445	1,3264	1,3165	1,3165	1,0496

Registros: 6139

Fallos durante el cálculo de Pk

Circuito	Estrategia	Tecnologia	Fallo_precio	Fallo_area	Fallo_fmax	Fallo_pot	Fallo_Tcpu	Fallo_Memo
ADCRECV	BALANCED	XA3S1200E-4F...	☑	☐	☐	☑	☐	☐
ADCRECV	BALANCED	XA3S1200E-4F...	☐	☐	☐	☑	☐	☐
ADCRECV	BALANCED	XA3S400A-4FG...	☑	☐	☐	☐	☐	☐
ADCRECV	BALANCED	XC3S100E-5CP...	☑	☐	☐	☐	☐	☐

Registros: 1244

Figura 56. Pantalla de presentación de las puntuaciones Pk y Pk' y de los fallos encontrados

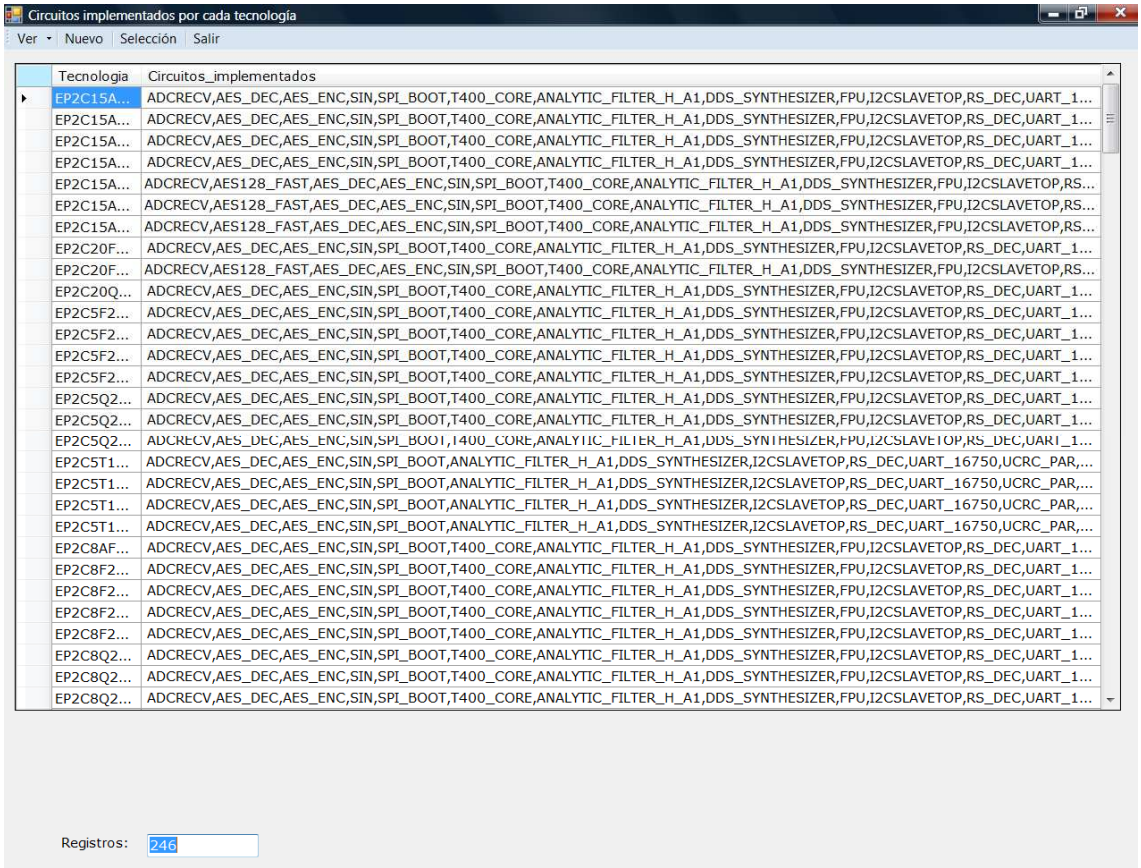
6.2.4. Tercera tarea

La aplicación analiza todos los datos leídos de los informes y en base a ellos genera un listado con todas las tecnologías objetivo del PdC acompañados por los circuitos implementadas con éxito en cada una de ellas. Esta relación queda almacenada en la tabla Circuitos_implementados y se presenta en una nueva pantalla tal y como puede verse en la Figura 57.

Por último, ofrece la posibilidad de seleccionar qué circuitos, se asignarán a cada sector comercial: automoción, electrónica de

consumo y comunicaciones, Figura 58. Nuevamente se desarrolla esta parte de la aplicación de un modo genérico para que se adapte a futuras modificaciones o ampliaciones del PdC, el conjunto de núcleos seleccionables dependerá del que la propia aplicación haya determinado según el apartado 6.2.3, almacenados en la tabla *Circuito.*

En este trabajo la selección se hace de entre los 16 circuitos que componen el PdC determinado en el capítulo 4, siguiendo la clasificación establecida en ese mismo capítulo, Figura 59.



Tecnologia	Circuitos_implementados
EP2C15A...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C15A...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C15A...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C15A...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C15A...	ADCRECV,AES128_FAST,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS...
EP2C15A...	ADCRECV,AES128_FAST,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS...
EP2C15A...	ADCRECV,AES128_FAST,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS...
EP2C20F...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C20F...	ADCRECV,AES128_FAST,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS...
EP2C20Q...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C5F2...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C5F2...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C5F2...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C5F2...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C5Q2...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C5Q2...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C5Q2...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C5T1...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,I2CSLAVETOP,RS_DEC,UART_16750,UCRC_PAR,...
EP2C5T1...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,I2CSLAVETOP,RS_DEC,UART_16750,UCRC_PAR,...
EP2C5T1...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,I2CSLAVETOP,RS_DEC,UART_16750,UCRC_PAR,...
EP2C5T1...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,I2CSLAVETOP,RS_DEC,UART_16750,UCRC_PAR,...
EP2C8AF...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C8F2...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C8F2...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C8F2...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C8F2...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C8Q2...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C8Q2...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...
EP2C8Q2...	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,UART_1...

Registros: 246

Figura 57. Pantalla de presentación de los circuitos del PdC implementados con éxito en cada FPGA

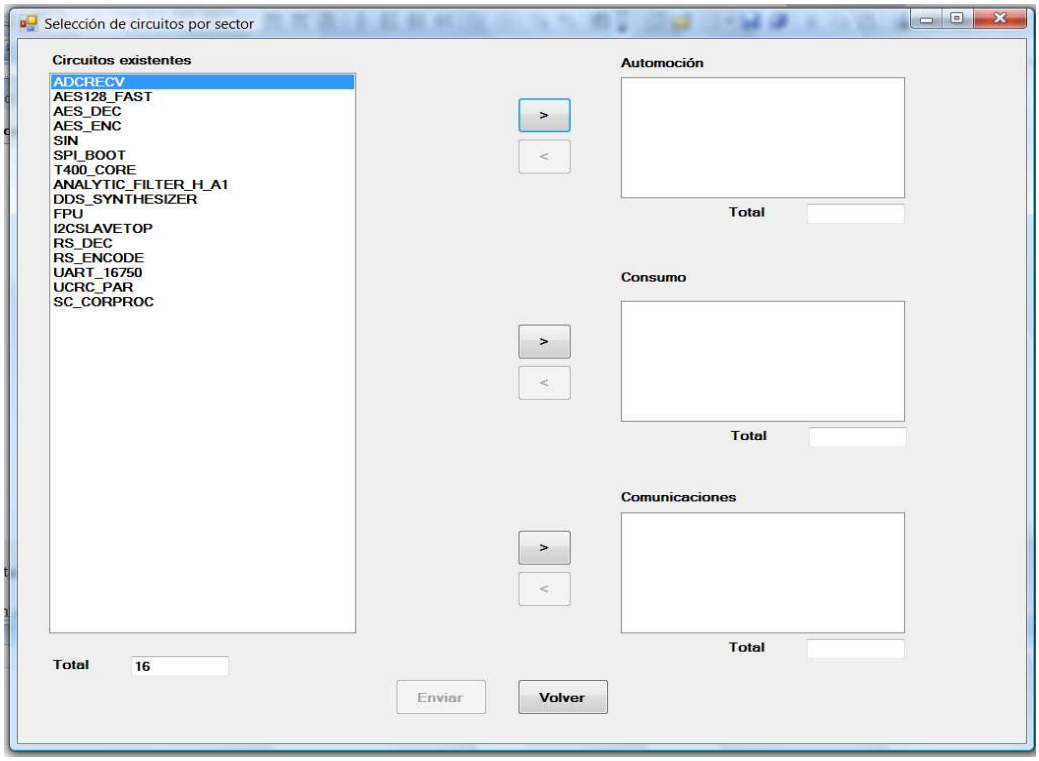


Figura 58. Pantalla de selección de los circuitos correspondientes a cada sector

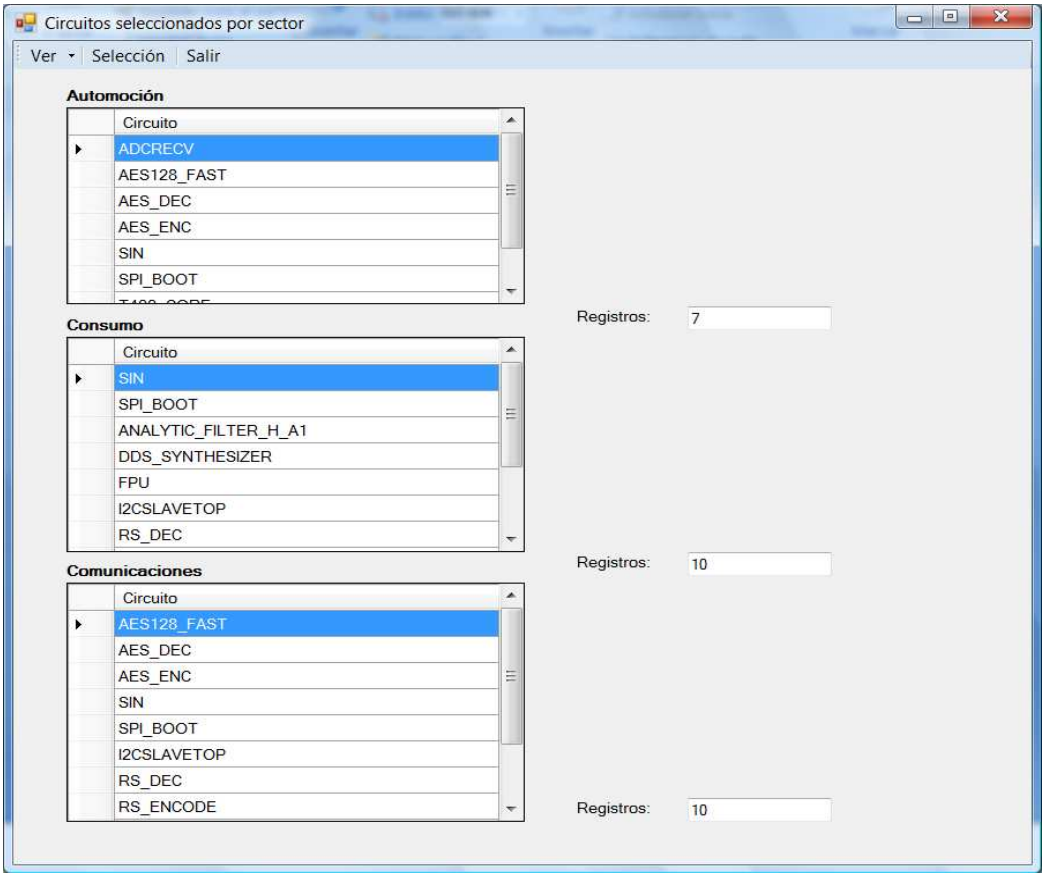


Figura 59. Núcleos del PdC asignados a cada sector comercial.

6.3. Aplicación 2: Asistente de clasificación de FPGAs

La aplicación desarrollada y descrita en el apartado anterior generó y almacenó toda la información necesaria para establecer una clasificación del conjunto de FPGAs tomadas como muestra.

El segundo programa ofrece al usuario la posibilidad de filtrar las FPGAs atendiendo a su precio y a algunos criterios constructivos para más tarde ordenar el subconjunto resultante según la puntuación obtenida por cada una en el PdC.

La pieza clave en este programa es una tabla de la BD denominada *Características,* Tabla 61, que deberá modificarse en el momento en que la muestra de FPGAs cambie. Si es necesario también habrá que actualizar el campo correspondiente a los precios de cada modelo de FPGA que se incluya en el estudio.

Todas las características y el precio de los modelos referidos en el capítulo 3 aparecen en esta tabla.

La tabla *Características* puede manipularse en el entorno de Access o bien utilizando el gestor de hojas de cálculo Excel para más tarde copiarla a la BD de Access.

6.3.1. Fase inicial de ejecución

Al ejecutar el programa, en primer lugar tiene lugar una lectura de todos los registros de la tabla *Características*. Toda la información extraída es tratada para eliminar duplicidades y para presentarla al usuario de forma ordenada.

6.3.2. Criterios de búsqueda

En esta primera versión es posible seleccionar FPGAs según los siguientes criterios, Figura 60:

- Precio
- Encapsulado
- Cantidad de unidades lógicas
- Cantidad de unidades de E/S
- Temperatura máxima
- Temperatura mínima

Por cada uno de estos parámetros se presenta una lista que recopila todos sus valores posibles conforme al conjunto de FPGAs analizadas y que aparecen almacenados en la tabla *Características*, habiendo sido leídos y tratados como ya se comentó en el apartado 6.3.1., Tabla 62.

Tras definir los criterios de filtrado del conjunto de FPGAs, hay que indicar qué circuitos participarán en la puntuación eligiendo entre las siguientes opciones:

- Todos los circuitos del PdC
- Conjunto de circuitos asignados al sector automoción[28]
- Conjunto de circuitos asignados al sector consumo

[28] Esta asignación debe haberse efectúado durante la ejecución de la aplicación ResultadosPdC

- Conjunto de circuitos asignados al sector comunicaciones

Por último se especificará el método para obtener las puntuaciones, a saber:

- Calcular la puntuación de aquella FPGAs filtradas en las que se ha conseguido implementar todos los circuitos marcados en el paso anterior.
- Calcular la puntuación de todas las FPGAs filtradas utilizando solo los circuitos marcados implementados en común.
- Seleccionar los circuitos, del total del PdC, en base a los cuales se puntuará cada FPGA filtrada. Este camino se resolverá de dos formas posibles:
    - Calculando la puntuación de todas las FPGAs seleccionadas con los circuitos implementados en común.
    - Si ningún circuito es común a todas las FPGAs, se puntúa solo el subconjunto formado por las que recibieron adecuadamente todos los circuitos.

La Tabla 60 resume las combinaciones posibles:

CIRCUITOS		PUNTUACION
Total PdC	**Todos los circuitos**	Solo FPGAs con todos los circuitos implementados
		Con los circuitos implementados en todas las FPGAs

CIRCUITOS		PUNTUACION
	Circuitos seleccionados[29]	Con los circuitos implementados en todas las FPGAs
		Solo FPGAs con todos los circuitos implementados
PdC por sector		Solo FPGAs con todos los circuitos implementados
		Con los circuitos implementados en todas las FPGAs

Tabla 60. Ordenación de los resultados posibles

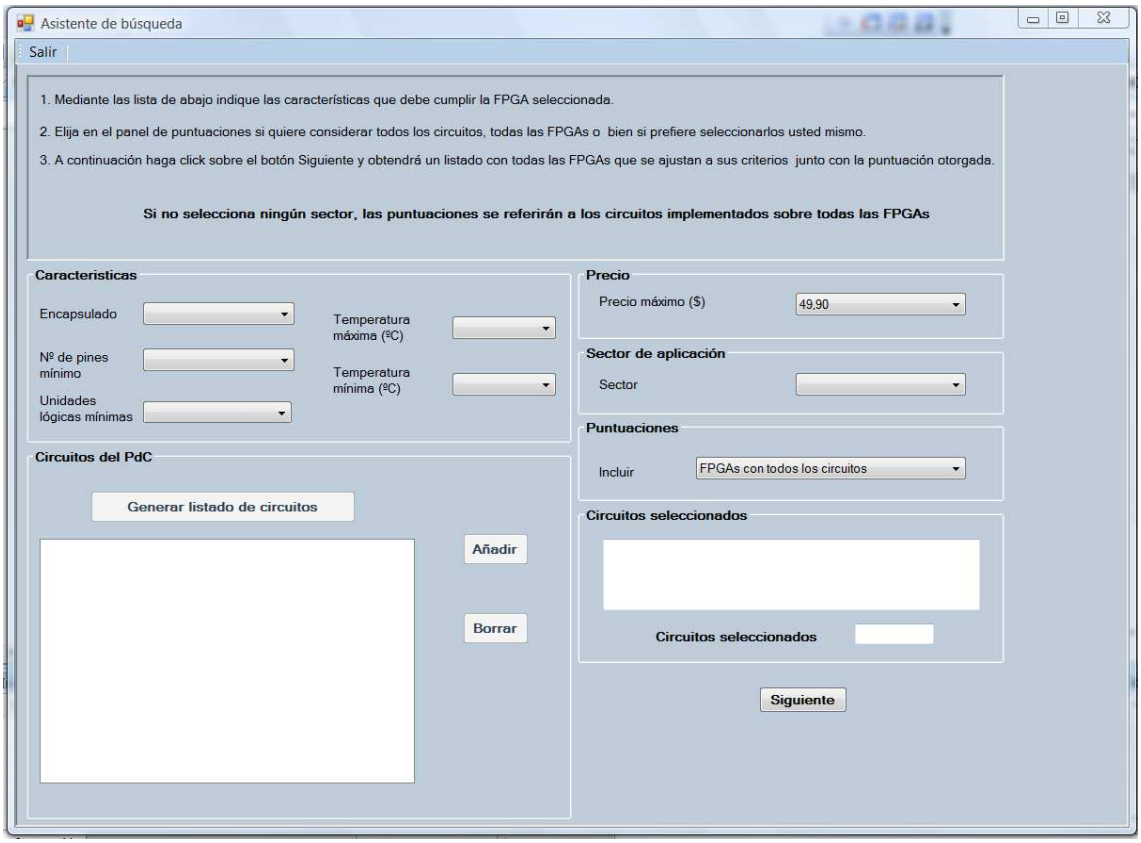


Figura 60.Pantalla del asistente de selección de FPGAs

[29] Si el camino que toma no es seleccionable, la aplicación intenta la primera opción y solo si no lo consigue toma la otra vía.

Modelo	Precio	Id	Encapsulado	Puertas	LUTs_FF	Fint_MHz	Tension_V	Pines_utiles	RAM_bits	Flash_bits	Mult_18x18	Ges_clk	T_MAX	T_MIN
XC3S50A-4VQ100C	5,52	229	100VTQFP	50000	1408	667,00	1,20	68	55296	0	3	2	85	0
XC3S50A-4VQ100I	6,36	230	100VTQFP	50000	1408	667,00	1,20	68	55296	0	3	2	100	-40
XC3S50A-5VQ100C	6,60	233	100VTQFP	50000	1408	770,00	1,20	68	55296	0	3	2	85	0
XC3S50A-4TQ144C	7,44	227	144TQFP EP	50000	1408	667,00	1,20	108	55296	0	3	2	85	0
XC3S50A-4TQ144I	8,58	228	144TQFP EP	50000	1408	667,00	1,20	108	55296	0	3	2	100	-40
XC3S50A-4FT256C	8,76	225	256FTBGA	50000	1408	667,00	1,20	144	55296	0	3	2	85	0
XC3S50A-5TQ144C	8,94	232	144TQFP EP	50000	1408	770,00	1,20	108	55296	0	3	2	85	0
XC3S50AN-4TQG144C	8,94	234	144TQFP EP	50000	1408	667,00	1,20	108	55296	1	3	2	85	0
XC3S50-4CP132C	9,30	213	132CSBGA	50000	1536	630,00	1,20	89	73728	0	50	2	85	0
XC3S50-4TQ144C	9,30	217	144TQFP	50000	1536	630,00	1,20	97	73728	0	50	2	85	0
XC3S100E-4VQ100C	9,48	120	100VTQFP	100000	1920	572,00	1,20	66	73728	0	4	2	85	0
XC3S50A-4FT256I	10,08	226	256FTBGA	50000	1408	667,00	1,20	144	55296	0	3	2	100	-40
XC3S50-4VQ100C	10,10	219	100VTQFP	50000	1536	630,00	1,20	63	73728	0	50	2	85	0
XC3S50-4VQ100I	10,10	220	100VTQFP	50000	1536	630,00	1,20	63	73728	0	50	2	100	-40
XC3S50-5VQ100C	10,10	224	100VTQFP	50000	1536	725,00	1,20	63	73728	0	50	2	85	0
XC3S50AN-4TQG144I	10,32	235	144TQFP EP	50000	1408	667,00	1,20	108	55296	1	3	2	100	-40
XC3S50A-5FT256C	10,56	231	256FTBGA	50000	1408	770,00	1,20	144	55296	0	3	2	85	0
XC3S50-4CP132I	10,70	214	132CSBGA	50000	1536	630,00	1,20	89	73728	0	50	2	100	-40
XC3S50-5CP132C	10,70	221	132CSBGA	50000	1536	725,00	1,20	89	73728	0	50	2	85	0
XC3S50-4TQ144I	10,70	218	144TQFP	50000	1536	630,00	1,20	97	73728	0	50	2	100	-40
XC3S50-5TQ144C	10,70	223	144TQFP	50000	1536	725,00	1,20	97	73728	0	50	2	85	0
XC3S50AN-5TQG144C	10,74	236	144TQFP EP	50000	1408	770,00	1,20	108	55296	1	3	2	85	0
XC3S100E-4VQ100I	10,92	121	100VTQFP	100000	1920	572,00	1,20	66	73728	0	4	2	100	-40
XC3S100E-5VQ100C	10,92	124	100VTQFP	100000	1920	657,00	1,20	66	73728	0	4	2	85	0
XC3S100E-4CP132C	11,10	116	132CSBGA	100000	1920	572,00	1,20	83	73728	0	4	2	85	0

Tabla 61.Ejemplo de algunos registros de la tabla *Características*

Temp_max	Temp_min	Encapsulado	Pines		Precio							ULS
				156	49,90	40,98	34,14	28,02	23,28	18,24	13,08	
85	-40	100VTQFP	63	158	49,30	40,50	33,78	27,42	23,10	18,18	12,84	1408
100	0	132CSBGA	66	160	48,48	39,90	33,60	27,00	22,50	18,12	12,80	1536
125		144-EQFP	68	161	48,30	39,50	33,20	26,82	22,44	17,90	12,78	1920
		144TQFP	82	168	48,06	39,30	33,12	26,80	22,40	17,58	12,10	3584
		144TQFP	83	172	47,50	38,90	32,16	26,70	21,99	16,70	12,06	3840
		144TQFP EP	84	173	47,18	38,88	32,00	26,40	21,95	16,68	11,70	4608
		164-MBGA	85	182	47,05	38,82	31,70	26,28	21,90	16,50	11,60	4896
		208PQFP	89	190	46,90	38,70	31,56	25,86	21,42	16,26	11,10	5136
		240-PQFP	92	195	46,70	38,04	30,90	25,80	21,06	16,14	10,92	7168
		256-BGA	94	215	46,60	38,00	30,36	25,55	20,76	15,84	10,74	8256
		256-FBGA	97	221	44,80	37,14	30,18	25,55	20,30	15,78	10,70	8320
		256FTBGA	106	232	44,58	36,96	29,64	25,45	20,25	15,42	10,70	9312
		256-UBGA	108	248	44,40	36,50	29,60	25,20	20,22	15,30	10,56	10320
		320FBGA	124	250	43,90	36,48	29,58	25,14	19,70	15,10	10,32	11776
		324-FBGA	138	251	43,74	36,30	29,22	24,90	19,50	15,06	10,10	14448
		400FBGA	141	264	43,00	35,70	29,10	24,36	19,20	14,90	10,08	15360
		400FPBGA	142	311	42,72	35,50	28,92	24,30	19,10	14,10	9,48	15408
		456FBGA	144	315	42,70	35,20	28,80	23,88	19,02	13,90	9,30	17344
		484FBGA	148	346	41,60	34,92	28,62	23,50	18,96	13,86	8,94	18752
		484UBGA	152	372	41,05	34,70	28,50	23,38	18,66	13,45	8,76	22528
						34,30	28,10	23,34	18,30	13,44	8,58	24624
											7,44	
											6,60	
											6,36	
											5,52	

Tabla 62. Valores de cada ítem para la muestra utilizada de FPGAs

6.3.3. Presentación de las puntuaciones Pk' y cálculo de PG

Una vez que se han seleccionado las caracteristicas, los circuitos y el método de puntuación, por ejemplo como aparece en la Figura 61, se muestra una pantalla con todas las puntuaciones Pk'y PG para el correspondiente subconjunto de FPGAs.

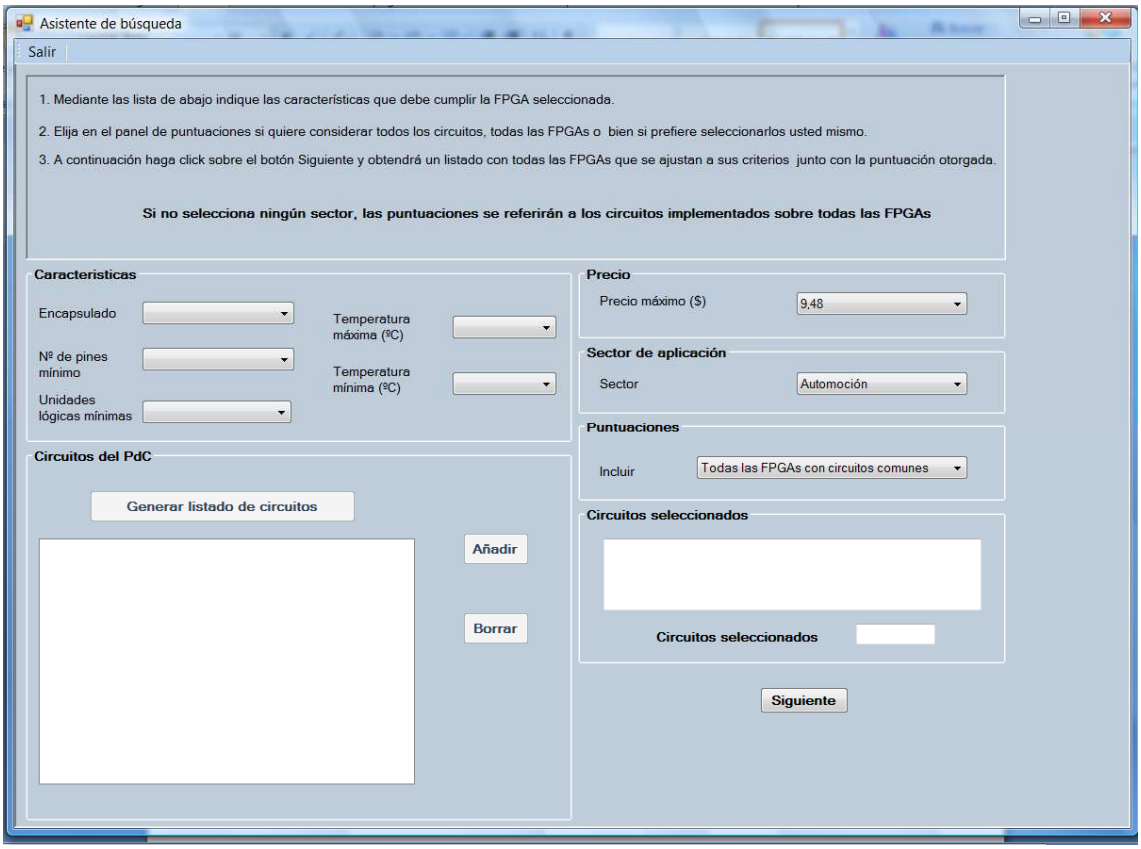
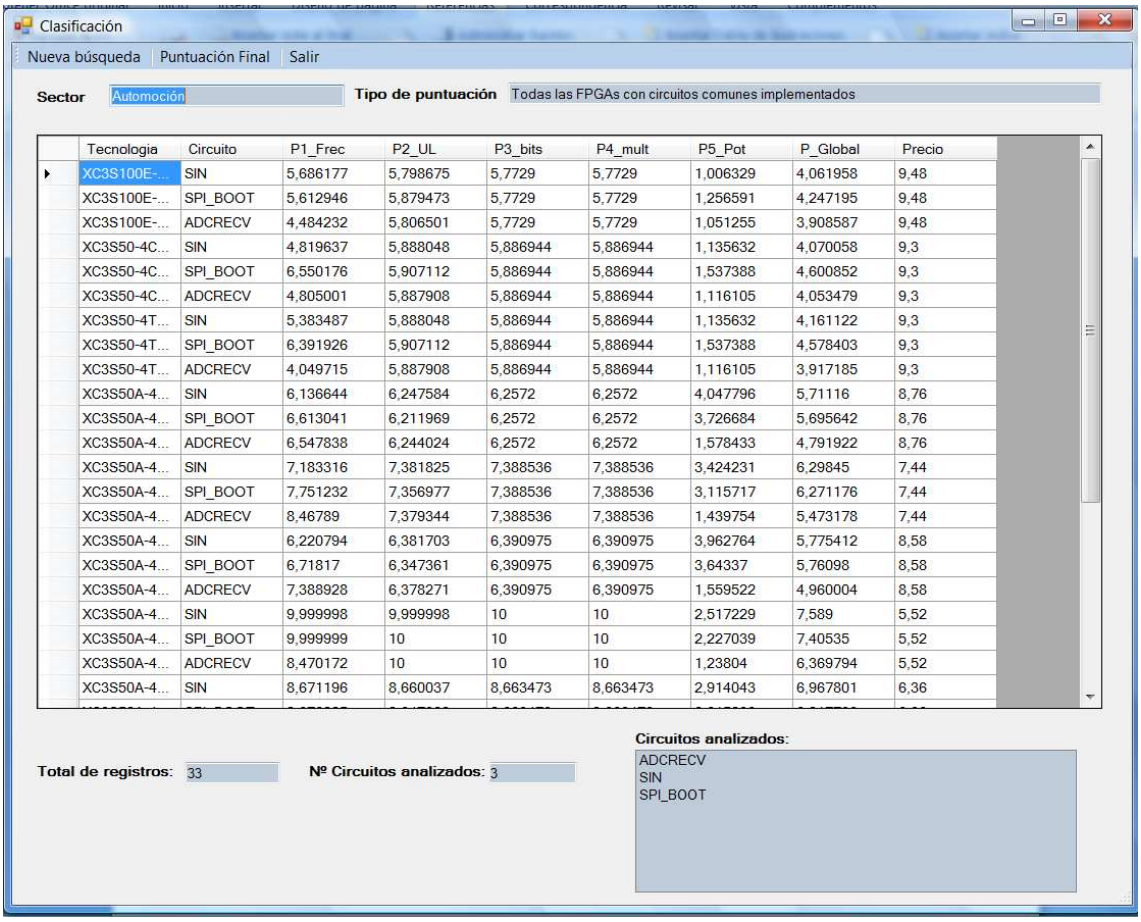


Figura 61. Ejemplo de búsqueda de FPGAs con un precio máximo de 9,48 $ puntuados conforme a los circuitos del sector automoción. Sólo circuitos comunes a todas las FPGAs.

Se comprueba en la Figura 62, como la aplicación devuelve información sobre cada FPGA seleccionada para cada uno de los circuitos fructuosos. Por cada combinación FPGA-circuito muestra los valores calculados de las puntuaciones normalizadas Pk', la puntuación global que evalúa su comportamiento medio ante todos los parámetros y como referencia el precio de cada tecnología.

Junto a los datos de puntuación aparecen otros elementos informativos que recuerdan al usuario los detalles de su búsqueda, como el sector y método de puntución, asi como el total de resultados obtenidos y los circuitos incluídos en los cálculos.

Es posible ordenar de mayor a menor y viceversa cada una de las columnas que componen la tabla protagonista de esta ventana. Es una ayuda para establecer una clasificación atendiendo a una sola de las Pk', PG o precio.



Tecnologia	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
XC3S100E-...	SIN	5,686177	5,798675	5,7729	5,7729	1,006329	4,061958	9,48
XC3S100E-...	SPI_BOOT	5,612946	5,879473	5,7729	5,7729	1,256591	4,247195	9,48
XC3S100E-...	ADCRECV	4,484232	5,806501	5,7729	5,7729	1,051255	3,908587	9,48
XC3S50-4C...	SIN	4,819637	5,888048	5,886944	5,886944	1,135632	4,070058	9,3
XC3S50-4C...	SPI_BOOT	6,550176	5,907112	5,886944	5,886944	1,537388	4,600852	9,3
XC3S50-4C...	ADCRECV	4,805001	5,887908	5,886944	5,886944	1,116105	4,053479	9,3
XC3S50-4T...	SIN	5,383487	5,888048	5,886944	5,886944	1,135632	4,161122	9,3
XC3S50-4T...	SPI_BOOT	6,391926	5,907112	5,886944	5,886944	1,537388	4,578403	9,3
XC3S50-4T...	ADCRECV	4,049715	5,887908	5,886944	5,886944	1,116105	3,917185	9,3
XC3S50A-4...	SIN	6,136644	6,247584	6,2572	6,2572	4,047796	5,71116	8,76
XC3S50A-4...	SPI_BOOT	6,613041	6,211969	6,2572	6,2572	3,726684	5,695642	8,76
XC3S50A-4...	ADCRECV	6,547838	6,244024	6,2572	6,2572	1,578433	4,791922	8,76
XC3S50A-4...	SIN	7,183316	7,381825	7,388536	7,388536	3,424231	6,29845	7,44
XC3S50A-4...	SPI_BOOT	7,751232	7,356977	7,388536	7,388536	3,115717	6,271176	7,44
XC3S50A-4...	ADCRECV	8,46789	7,379344	7,388536	7,388536	1,439754	5,473178	7,44
XC3S50A-4...	SIN	6,220794	6,381703	6,390975	6,390975	3,962764	5,775412	8,58
XC3S50A-4...	SPI_BOOT	6,71817	6,347361	6,390975	6,390975	3,64337	5,76098	8,58
XC3S50A-4...	ADCRECV	7,388928	6,378271	6,390975	6,390975	1,559522	4,960004	8,58
XC3S50A-4...	SIN	9,999998	9,999998	10	10	2,517229	7,589	5,52
XC3S50A-4...	SPI_BOOT	9,999999	10	10	10	2,227039	7,40535	5,52
XC3S50A-4...	ADCRECV	8,470172	10	10	10	1,23804	6,369794	5,52
XC3S50A-4...	SIN	8,671196	8,660037	8,663473	8,663473	2,914043	6,967801	6,36

Figura 62. Pantalla con las puntuacions Pk' y PG

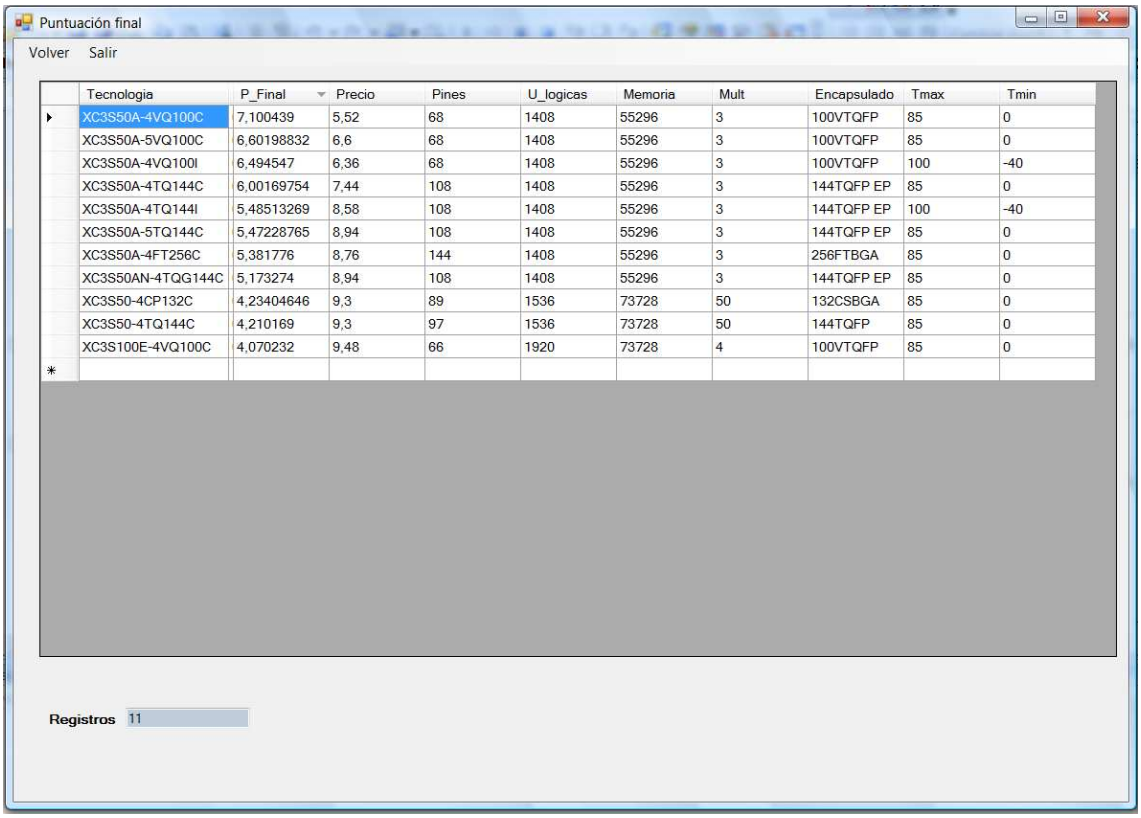
Desde esta ventana es posible comenzar una nueva búsqueda o bien avanzar para calcular las PF de las FPGAs filtradas, amén del botón de salida de la aplicación.

6.3.4. Clasificación final

Si se ha optado por avanzar hacia las puntuaciones finales aparecerá una ventana como la de la Figura 63, las once FPGAs que respondían a los criterios vistos en la Figura 61, se ordenan descendentemente según la PF resuelta para cada una de ellas según los circuitos mostrados en la Figura 62.

A cada tecnología y puntuación le acompañan algunas de sus características constructivas, en concreto aquellas que sirvieron como filtro de búsqueda. Como en la pantalla anterior, es posible ordenar según todas las columnas de la tabla.

Desde este punto, es posible volver a la ventana anterior desde la cual se podrá iniciar una nueva búsqueda, o bien salir de la aplicación.



Tecnologia	P_Final	Precio	Pines	U_logicas	Memoria	Mult	Encapsulado	Tmax	Tmin
XC3S50A-4VQ100C	7,100439	5,52	68	1408	55296	3	100VTQFP	85	0
XC3S50A-5VQ100C	6,60198832	6,6	68	1408	55296	3	100VTQFP	85	0
XC3S50A-4VQ100I	6,494547	6,36	68	1408	55296	3	100VTQFP	100	-40
XC3S50A-4TQ144C	6,00169754	7,44	108	1408	55296	3	144TQFP EP	85	0
XC3S50A-4TQ144I	5,48513269	8,58	108	1408	55296	3	144TQFP EP	100	-40
XC3S50A-5TQ144C	5,47228765	8,94	108	1408	55296	3	144TQFP EP	85	0
XC3S50A-4FT256C	5,381776	8,76	144	1408	55296	3	256FTBGA	85	0
XC3S50AN-4TQG144C	5,173274	8,94	108	1408	55296	3	144TQFP EP	85	0
XC3S50-4CP132C	4,23404646	9,3	89	1536	73728	50	132CSBGA	85	0
XC3S50-4TQ144C	4,210169	9,3	97	1536	73728	50	144TQFP	85	0
XC3S100E-4VQ100C	4,070232	9,48	66	1920	73728	4	100VTQFP	85	0

Figura 63. Pantalla con clasificación final basada en PF

6.4. Consideraciones

Las herramientas expuestas son útiles para satisfacer el objetivo f) citado en el capítulo 1. Este objetivo pretendía automatizar las tareas de ejecución del PdC, la lectura de los resultados devueltos por las herramientas, el tratamiento de los datos obtenidos, el cálculo de las puntuaciones conseguidas por cada FPGA y en base a ellas clasificarlas y por último ofrecer un asistente que permitiese seleccionar en base a parámetros constructivos las FPGAs con mejor puntuación.

Las herramientas nos permiten abordar la fase experimental que se presenta en el siguiente capítulo, y a partir de los resultados, establecer las conclusiones oportunas sobre las cuestiones planteadas inicialmente e incluso sobre otros temas no planteados.

7. Resultados experimentales

En este capítulo se muestran los resultados obtenidos tras sintetizar, implementar y analizar los 16 circuitos que integran el PdC propuesto en el capítulo 4. La ejecución se ha llevado a cabo utilizando tecnologías de Xilinx y Altera al ser estos fabricantes los que ofrecen herramientas gratuitas sin límite temporal. En total se han utilizado 186 tecnologias, 80 del fabricante Altera y 106 de Xilinx, el listado completo se encuentra referido en el apartado 3.5 , en el cual el total de Xilinx es de 167 ya que incluye las dos variantes posibles para cada modelo, industrial y comercial, siempre y cuando su precio por unidad sea inferior a 50$.

Todos los resultados han sido obtenidos en las siguientes condiciones, Tabla 63 :

	Xilinx	**Altera**
Herramienta	Webpack Ise 10.1	Quartus II 9.0 web edition
Estrategias	Balanced/Timing Performance	Balanced/Speed
Sistemas Operativo	Windows Vista™ Home Premium (32 bits)	
Equipo informático	Intel Core2 Quad Q6600 @2,4GHz Memoria 4Gbytes PC2-5300 DDR2	

Tabla 63. Condiciones de ejecución del PdC

7.1. Ejecución del PdC

La ejecución del PdC se realizó siguiendo todas las instrucciones indicadas en el apartado 4.4.1., iniciando a

continuación los ficheros por lotes aportados en 4.4.2. En primer lugar se utilizó como argumento el parámetro de estrategia equilibrada, “balanced” y a su término se repitió la operación esta vez con el argumento “tiempo”.

Dado el número de circuitos y tecnologías que se utilizan como muestra, este trabajo necesitó de un tiempo considerable para llegar a su fin y de una gran cantidad de memoria de almacentamiento, Tabla 64.

80 FPGAs y 16 circuitos		**106 FPGAs y 16 circuitos**		**TOTAL**
Altera Balanced	**Altera Tiempo**	**Xilinx Balanced**	**Xilinx Tiempo**	
21 horas y 50 minutos	20 horas y 15 minutos	89 horas y 10 minutos	68 horas	≈ 200 horas ó 8 dias y 8 horas
Tiempo promedio por circuito y tecnología				
1 minuto y 12 segundos	57 segundos	3 minutos y 9 segundos	2 minutos y 24 segundos	

MB ocupados por Resultados Altera		**MB ocupados por Resultados Xilinx**	
Balanced	Tiempo	Balanced	Tiempo
8,39 GB	8,45 GB	8,11 GB	8,13 GB

Tabla 64. Estimación del tiempo empleado por el equipo informático para finalizar el PdC

La ejecución del PdC generó un gran número de ficheros siendo de interés los que se relacionan en la Tabla 65, junto a ellos se indica el número total de informes que la aplicación ***ResultadosPdC*** abrió para obtener todos los parámetros relevantes.

	Xilinx	**Altera**
Núcleos PdC	16	
Tecnologías	106	80
Estrategias	Balanced / Tiempo	

	Xilinx	Altera
Fichero Síntesis	1.696	-
Fichero Mapeado	1.696	-
Fichero Emplazamiento y ruteado	1.696	-
Fichero global	-	1.280
Fichero de tiempo	1.696	1.280
Fichero de Potencia	1.696	1.280
TOTAL FICHEROS[30]	**8.480/16.960**	**3.840/7.680**

Tabla 65. Ficheros de interés generados tras la ejecución del PdC

7.2. Lectura de resultados

Tras la ejecución del PdC se inició la aplicación ***ResultadosPdC*** para proceder a la lectura de los ficheros indicados en Tabla 65 con el objetivo de extraer la información importante. Para conseguirlo se eligió *NuevaLectura* del menú de la pantalla principal del programa y en el cuadro de diálogo se especificó la ruta donde debía localizar los resultados, carpetas *Resultados_Altera* y *Resultados_Xilinx,* ubicadas en el disco c:\ del equipo informático.

Tres son las tablas de la BD, que presenta la aplicación en su primera pantalla: *DatosReports, Fallos* y *Sobredimensionados.* A la vista de los datos barajados en este trabajo, Tabla 65, se esperaban un total de 5.952 registros distribuidos como se refiere en la Tabla 66.

	Xilinx	Altera
Núcleos PdC	16	

[30] El total se duplica al haberse ejecutado el PdC con dos estrategias: Balanced y Tiempo

	Xilinx	Altera
Tecnologías	106	80
Estrategias	Balanced / Tiempo	
Nº Resultados esperados (Ci-Ti)	1.696 / 1.696	1.280 / 1.280
Nº Resultados leídos	1.297 / 1.305	1.095 / 1.084
Fallos de ejecución	0 / 0	10 / 10
Fallos de implementación	399 / 391	175 / 186

Tabla 66. Registro esperados y encontrados en las tablas DatosReports, Fallos y Sobredimensionados

Un análisis cuantitativo de la Tabla 66 revela una total correlación entre registros esperados y obtenidos.

7.2.1. Datos válidos: Tabla DatosReports

En la Tabla 68 se muestran algunos de los registros de la tabla *DatosReports* obtenida por la aplicación ***ResultadosPdC***. La cantidad total de registros que la integran, 4.780, hace inviable que se incluya en este documento, se ha optado por almacenar la BD al completo en un CD adjunto.

La parte de la tabla *DatosReports* aquí incluida recoge toda la información extraída para uno de los circuitos del PdC, en concreto el denominado ADCRECV. Los valores de los campos corresponden a los leídos tras la implementación de este núcleo sobre las 186 tecnologías de la muestra inicial con estrategia *Balanced (B)*. A partir de estos valores se calcularán las magnitudes que más adelante servirán para puntuar cada una de las tecnologías.

En la Tabla 67 se establece la correspondencia del encabezado de cada columna con su significado.

Nombre	Significado
UL_u[31]	Unidades lógicas utilizadas para implementar el circuito
UL_d	Unidades lógicas aún disponibles tras implementar el circuito
Bits_u	Bits correspondientes a los bloques de memoria utilizados para implementar el circuito
Bits_d	Bits correspondientes a los bloques de memoria aún disponibles tras implementar el circuito
Mult_u	Bloques multiplicadores utilizados para implementar el circuito
Mult_d	Bloques multiplicadores disponibles tras implementar el circuito
Pines_u	Pines de usuario utilizados para implementar el circuito
Pines_d	Pines disponibles por el usuarios restantes tras implementar el circuito
Potencia	Potencia de consumo aproximada del núcleo ADCRECV sobre cada tecnología que se ha implementado (mW)
Reg2Reg	Tiempo máximo de retardo entre dos FF del circuito implementado. Incluye el tiempo de propagación del FF origen, el retardo combinacional de los elementos situados entre el origen y el destino, de las conexiones y de establecimiento del dato en el FF destino (ns)
Tsu	Tiempo de establecimiento del dato externo (ns)
Th	Tiempo de mantenimiento del dato externo (ns)
Tco	Tiempo que tarda el dato en estar disponible en un terminal de salida tras llegar la señal de reloj al último FF del camino.
T_sint	Tiempo de CPU empleado por la herramienta de síntesis (s)
T_imp	Tiempo de CPU empleado por la herramienta de implementación (s)
Memo	Pico de memoria utilizado por alguna de las herramientas de cada entorno de trabajo (MB).

Tabla 67. Significado de las columnas de la Tabla 68

[31] Las unidades lógicas corresponden a *slices* para las tecnologías de Xilinx y a *logic elements* para las de Altera.

Circuito	Est.	Tecnologia	UL_u	UL_d	Bits_u	Bits_d	Mult_u	Mult_d	Pines_u	Pines_d	Potencia	Reg2Reg	Tsu	Th	Tco	T_sint	T_imp	Memo
ADCRECV	B	EP2C15AF256C6	38	14448	0	239616	0	52	27	152	73	2,754	4,95	-0,65	9,304	1	4	249
ADCRECV	B	EP2C15AF256C7	38	14448	0	239616	0	52	27	152	72	3,407	6,8	-0,5	10,43	1	4	249
ADCRECV	B	EP2C15AF256C8	38	14448	0	239616	0	52	27	152	72	3,994	5,7	-1,44	10,6	1	4	249
ADCRECV	B	EP2C15AF256I8	38	14448	0	239616	0	52	27	152	72	3,994	5,7	-1,44	10,6	1	4	249
ADCRECV	B	EP2C15AF484C7	38	14448	0	239616	0	52	27	315	72	3,33	6,13	-1,06	10,29	1	4	249
ADCRECV	B	EP2C15AF484C8	38	14448	0	239616	0	52	27	315	72	3,987	6,84	-1,5	11,72	1	4	249
ADCRECV	B	EP2C15AF484I8	38	14448	0	239616	0	52	27	315	72	3,987	6,84	-1,5	11,72	1	4	249
ADCRECV	B	EP2C20F256C8	38	18752	0	239616	0	52	27	152	72	4,018	6,33	-0,8	13,08	1	4	249
ADCRECV	B	EP2C20F484C8	38	18752	0	239616	0	52	27	315	72	4,016	6,31	-0,88	11,72	1	4	249
ADCRECV	B	EP2C20Q240C8	38	18752	0	239616	0	52	27	142	72	3,995	6,26	-1,49	10,95	1	4	249
ADCRECV	B	EP2C5F256C6	38	4608	0	119808	0	26	27	158	34	2,84	4,36	-0,82	8,243	1	2	220
ADCRECV	B	EP2C5F256C7	38	4608	0	119808	0	26	27	158	34	3,296	4,75	-0,89	9,243	1	2	220
ADCRECV	B	EP2C5F256C8	38	4608	0	119808	0	26	27	158	34	3,833	5,17	-1,14	10,02	1	2	220
ADCRECV	B	EP2C5F256I8	38	4608	0	119808	0	26	27	158	34	3,833	5,17	-1,14	10,02	1	2	220
ADCRECV	B	EP2C5Q208C7	38	4608	0	119808	0	26	27	142	34	3,669	3,75	-0,38	9,784	1	2	220
ADCRECV	B	EP2C5Q208C8	38	4608	0	119808	0	26	27	142	34	4,077	5,28	-0,79	10,8	1	2	220
ADCRECV	B	EP2C5Q208I8	38	4608	0	119808	0	26	27	142	34	4,077	5,28	-0,79	10,8	1	2	220
ADCRECV	B	EP2C5T144C6	38	4608	0	119808	0	26	27	89	35	3,072	5,13	-0,59	8,096	1	2	220
ADCRECV	B	EP2C5T144C7	38	4608	0	119808	0	26	27	89	35	3,36	4,95	-1,29	9,083	1	2	220
ADCRECV	B	EP2C5T144C8	38	4608	0	119808	0	26	27	89	35	3,988	5,57	-1,44	10,25	1	2	220
ADCRECV	B	EP2C5T144I8	38	4608	0	119808	0	26	27	89	35	3,988	5,57	-1,44	10,25	1	2	220
ADCRECV	B	EP2C8AF256I8	38	8256	0	165888	0	36	27	182	46	4,015	6,37	-1,32	10,23	1	3	227
ADCRECV	B	EP2C8F256C6	38	8256	0	165888	0	36	27	182	46	2,858	5,16	-0,58	8,224	1	3	227
ADCRECV	B	EP2C8F256C7	38	8256	0	165888	0	36	27	182	46	3,568	5,94	-0,74	9,726	1	3	227
ADCRECV	B	EP2C8F256C8	38	8256	0	165888	0	36	27	182	46	4,015	6,37	-1,32	10,23	1	3	227
ADCRECV	B	EP2C8F256I8	38	8256	0	165888	0	36	27	182	46	4,015	6,37	-1,32	10,23	1	3	227
ADCRECV	B	EP2C8Q208C7	38	8256	0	165888	0	36	27	138	46	3,345	5,15	-0,33	9,277	1	3	227

Circuito	Est.	Tecnologia	UL_u	UL_d	Bits_u	Bits_d	Mult_u	Mult_d	Pines_u	Pines_d	Potencia	Reg2Reg	Tsu	Th	Tco	T_sint	T_imp	Memo
ADCRECV	B	EP2C8Q208C8	38	8256	0	165888	0	36	27	138	46	4,023	5,6	-0,25	10,47	1	3	227
ADCRECV	B	EP2C8Q208I8	38	8256	0	165888	0	36	27	138	46	4,023	5,6	-0,25	10,47	1	3	227
ADCRECV	B	EP2C8T144C6	38	8256	0	165888	0	36	27	85	46	2,987	5,51	-0,47	8,732	1	3	227
ADCRECV	B	EP2C8T144C7	38	8256	0	165888	0	36	27	85	46	3,817	6,1	-0,78	9,347	1	3	227
ADCRECV	B	EP2C8T144C8	38	8256	0	165888	0	36	27	85	46	4,072	6,43	-1,06	10,22	1	3	227
ADCRECV	B	EP2C8T144I8	38	8256	0	165888	0	36	27	85	46	4,072	6,43	-1,06	10,22	1	3	227
ADCRECV	B	EP3C10E144A7	38	10320	0	423936	0	46	27	95	59	3,729	4,33	-1,27	9,916	1	4	243
ADCRECV	B	EP3C10E144C7	38	10320	0	423936	0	46	27	95	59	3,579	4,18	-1,2	8,654	1	5	255
ADCRECV	B	EP3C10E144C8	38	10320	0	423936	0	46	27	95	59	4,137	4,71	-0,45	10,11	1	6	257
ADCRECV	B	EP3C10E144I7	38	10320	0	423936	0	46	27	95	59	3,61	4,24	-1,22	8,784	1	6	256
ADCRECV	B	EP3C10F256A7	38	10320	0	423936	0	46	27	183	59	3,777	4,13	-0,91	9,557	2	4	243
ADCRECV	B	EP3C10F256C6	38	10320	0	423936	0	46	27	183	59	3,143	3,42	-0,72	8,338	1	5	255
ADCRECV	B	EP3C10F256C7	38	10320	0	423936	0	46	27	183	59	3,65	3,95	-0,88	9,794	1	6	256
ADCRECV	B	EP3C10F256C8	38	10320	0	423936	0	46	27	183	59	4,277	4,38	-0,52	11,05	1	6	257
ADCRECV	B	EP3C10F256I7	38	10320	0	423936	0	46	27	183	59	3,681	4,01	-0,89	9,917	1	6	262
ADCRECV	B	EP3C10U256C6	38	10320	0	423936	0	46	27	183	59	3,158	3,42	-0,72	8,333	1	5	255
ADCRECV	B	EP3C10U256C7	38	10320	0	423936	0	46	27	183	59	3,676	3,92	-0,96	9,233	1	5	257
ADCRECV	B	EP3C10U256C8	38	10320	0	423936	0	46	27	183	59	4,277	4,39	-0,52	11,03	1	6	257
ADCRECV	B	EP3C10U256I7	38	10320	0	423936	0	46	27	183	59	3,705	3,88	-0,91	9,354	1	6	262
ADCRECV	B	EP3C16E144C7	38	15408	0	516096	0	112	27	85	68	3,901	3,23	-0,51	9,5	1	6	259
ADCRECV	B	EP3C16E144C8	38	15408	0	516096	0	112	27	85	68	4,677	3,64	-0,61	11,53	1	6	261
ADCRECV	B	EP3C16F256C6	38	15408	0	516096	0	112	27	169	68	3,302	3,78	-0,99	7,925	1	6	259
ADCRECV	B	EP3C16F256C7	38	15408	0	516096	0	112	27	169	68	3,808	3,76	-0,84	9,361	1	6	261
ADCRECV	B	EP3C16F256C8	38	15408	0	516096	0	112	27	169	68	4,278	4,32	-1,08	10,35	1	6	261
ADCRECV	B	EP3C16F484C6	38	15408	0	516096	0	112	27	347	68	3,243	2,93	0,108	9,247	1	6	259
ADCRECV	B	EP3C16F484C7	38	15408	0	516096	0	112	27	347	68	3,822	4,25	-0,66	10,22	1	6	261
ADCRECV	B	EP3C16F484C8	38	15408	0	516096	0	112	27	347	68	3,905	3,57	-0,55	9,279	1	6	261

Circuito	Est.	Tecnologia	UL_u	UL_d	Bits_u	Bits_d	Mult_u	Mult_d	Pines_u	Pines_d	Potencia	Reg2Reg	Tsu	Th	Tco	T_sint	T_imp	Memo
ADCRECV	B	EP3C16M164C7	38	15408	0	516096	0	112	27	93	68	3,627	3,45	-0,6	9,077	1	6	259
ADCRECV	B	EP3C16M164C8	38	15408	0	516096	0	112	27	93	68	3,976	3,76	-0,74	10,79	1	6	261
ADCRECV	B	EP3C16Q240C8	38	15408	0	516096	0	112	27	161	68	4,157	4,14	-0,69	11,96	1	6	260
ADCRECV	B	EP3C16U256C6	38	15408	0	516096	0	112	27	169	68	3,171	3,66	-0,98	7,939	1	6	259
ADCRECV	B	EP3C16U256C7	38	15408	0	516096	0	112	27	169	68	3,808	3,78	-0,84	9,361	1	6	261
ADCRECV	B	EP3C16U256C8	38	15408	0	516096	0	112	27	169	68	4,284	4,34	-1,08	10,29	1	6	261
ADCRECV	B	EP3C16U484C7	38	15408	0	516096	0	112	27	347	68	3,821	4,01	-0,66	9,488	1	6	261
ADCRECV	B	EP3C16U484C8	38	15408	0	516096	0	112	27	347	68	3,902	3,58	-0,52	9,223	1	6	261
ADCRECV	B	EP3C25E144C8	38	24624	0	608256	0	132	27	83	95	4,08	4,14	-0,86	11,49	1	8	265
ADCRECV	B	EP3C25F256C8	38	24624	0	608256	0	132	27	157	95	3,91	3,96	-0,52	11,68	1	8	265
ADCRECV	B	EP3C25F324C8	38	24624	0	608256	0	132	27	216	95	3,912	3,93	-1,27	12,67	1	8	265
ADCRECV	B	EP3C25Q240C8	38	24624	0	608256	0	132	27	149	95	3,994	4,22	-0,97	12,56	1	7	263
ADCRECV	B	EP3C25U256C8	38	24624	0	608256	0	132	27	157	95	3,91	3,95	-0,52	11,67	1	8	265
ADCRECV	B	EP3C5E144A7	38	5136	0	423936	0	46	27	95	59	3,555	4,19	-1,3	8,482	2	4	243
ADCRECV	B	EP3C5E144C7	38	5136	0	423936	0	46	27	95	59	3,635	4	-1,26	8,748	1	6	255
ADCRECV	B	EP3C5E144C8	38	5136	0	423936	0	46	27	95	59	3,903	4,83	-1,13	9,276	1	6	257
ADCRECV	B	EP3C5E144I7	38	5136	0	423936	0	46	27	95	59	3,665	4,06	-1,28	8,885	1	6	256
ADCRECV	B	EP3C5F256C6	38	5136	0	423936	0	46	27	183	59	3,114	3,6	-1,06	7,83	1	5	255
ADCRECV	B	EP3C5F256C7	38	5136	0	423936	0	46	27	183	59	3,46	4,17	-1,24	8,717	1	6	257
ADCRECV	B	EP3C5F256C8	38	5136	0	423936	0	46	27	183	59	4,321	4,47	-0,76	9,473	1	6	257
ADCRECV	B	EP3C5F256I7	38	5136	0	423936	0	46	27	183	59	3,642	4,23	-1,22	9,125	1	6	262
ADCRECV	B	EP3C5M164C7	38	5136	0	423936	0	46	27	107	59	3,455	3,8	-0,83	7,785	1	6	255
ADCRECV	B	EP3C5U256C6	38	5136	0	423936	0	46	27	183	59	3,119	3,6	-1,06	7,8	1	6	255
ADCRECV	B	EP3C5U256C7	38	5136	0	423936	0	46	27	183	59	3,453	4,11	-1,32	9,113	1	6	257
ADCRECV	B	EP3C5U256C8	38	5136	0	423936	0	46	27	183	59	4,32	4,26	-0,78	10,04	1	6	257
ADCRECV	B	EP3C5U256I7	38	5136	0	423936	0	46	27	183	59	3,492	4,23	-1,25	8,801	1	6	262
ADCRECV	B	XA3S100E-4CPG132	21	960	0	0	0	0	27	83	3	4,601	3,96	1,096	9,468	3,18	4	164

Circuito	Est.	Tecnologia	UL_u	UL_d	Bits_u	Bits_d	Mult_u	Mult_d	Pines_u	Pines_d	Potencia	Reg2Reg	Tsu	Th	Tco	T_sint	T_imp	Memo
ADCRECV	B	XA3S100E-4TQG144	21	960	0	0	0	0	27	108	3	3,826	3,54	1,072	9,686	3,2	4	165
ADCRECV	B	XA3S100E-4VQG100	21	960	0	0	0	0	27	66	3	5,13	2,52	1,297	9,067	3,2	3	165
ADCRECV	B	XA3S1200E-4FTG256	21	8672	0	0	0	0	27	190	0	6,187	4,82	0,99	9,197	3,82	8	181
ADCRECV	B	XA3S200A-4FTG256	21	1792	0	0	0	0	27	195	27	5,601	4,01	0,545	9,465	3,6	6	180
ADCRECV	B	XA3S250E-4CPG132	21	2448	0	0	0	0	27	92	5	5,818	6,23	1,156	9,622	3,34	5	169
ADCRECV	B	XA3S250E-4FTG256	21	2448	0	0	0	0	27	172	5	4,908	2,91	1,215	9,953	3,34	5	169
ADCRECV	B	XA3S250E-4PQG208	21	2448	0	0	0	0	27	158	5	5,408	3,86	0,993	9,261	3,35	5	170
ADCRECV	B	XA3S250E-4TQG144	21	2448	0	0	0	0	27	108	5	5,734	4,43	0,888	9,83	3,32	5	169
ADCRECV	B	XA3S250E-4VQG100	21	2448	0	0	0	0	27	66	5	5,372	2,47	0,579	9,964	3,33	5	169
ADCRECV	B	XA3S400A-4FGG400	21	3584	0	0	0	0	27	311	31	3,686	3,97	0,886	8,924	3,81	7	183
ADCRECV	B	XA3S400A-4FTG256	21	3584	0	0	0	0	27	195	31	5,285	3,51	0,919	9,863	3,8	7	183
ADCRECV	B	XA3S500E-4CPG132	21	4656	0	0	0	0	27	92	8	5,39	4,46	1,021	8,853	3,51	6	173
ADCRECV	B	XA3S500E-4FTG256	21	4656	0	0	0	0	27	190	8	5,129	3,32	0,819	9,139	3,51	6	173
ADCRECV	B	XA3S500E-4PQG208	21	4656	0	0	0	0	27	158	8	5,388	3,6	1,464	9,807	3,48	6	173
ADCRECV	B	XC3S1000-4FG320	21	7680	0	0	0	0	27	221	103	5,975	3,06	1,144	9,932	3,22	4	170
ADCRECV	B	XC3S1000-4FT256	21	7680	0	0	0	0	27	173	103	5,45	3,88	0,9	9,537	3,24	4	169
ADCRECV	B	XC3S1000-5FT256	21	7680	0	0	0	0	27	173	103	4,951	3,12	0,738	8,497	3,22	4	169
ADCRECV	B	XC3S100E-4CP132	21	960	0	0	0	0	27	83	38	4,601	3,96	1,096	9,468	3,18	4	165
ADCRECV	B	XC3S100E-4TQ144	21	960	0	0	0	0	27	108	38	3,826	3,54	1,072	9,686	3,18	4	165
ADCRECV	B	XC3S100E-4VQ100	21	960	0	0	0	0	27	66	38	5,13	2,52	1,297	9,067	3,17	3	165
ADCRECV	B	XC3S100E-5CP132	21	960	0	0	0	0	27	83	38	5,314	3,05	1,363	9,012	3,19	4	165
ADCRECV	B	XC3S100E-5TQ144	21	960	0	0	0	0	27	108	38	3,181	2,37	1,122	8,695	3,19	4	165
ADCRECV	B	XC3S100E-5VQ100	21	960	0	0	0	0	27	66	38	4,313	2,26	1,317	8,437	3,2	4	165
ADCRECV	B	XC3S1200E-4FG320	21	8672	0	0	0	0	27	250	16	7,047	4,02	1,644	9,757	3,79	8	181
ADCRECV	B	XC3S1200E-4FT256	21	8672	0	0	0	0	27	190	16	6,187	4,82	0,99	9,197	3,79	8	181
ADCRECV	B	XC3S1200E-5FT256	21	8672	0	0	0	0	27	190	16	4,535	4,04	1,051	9,636	3,79	8	180
ADCRECV	B	XC3S1400A-4FT256	21	11264	0	0	0	0	27	161	62	5,041	2,38	0,765	9,304	4,46	10	196

Circuito	Est.	Tecnologia	UL_u	UL_d	Bits_u	Bits_d	Mult_u	Mult_d	Pines_u	Pines_d	Potencia	Reg2Reg	Tsu	Th	Tco	T_sint	T_imp	Memo
ADCRECV	B	XC3S1400A-5FT256	21	11264	0	0	0	0	27	161	62	3,384	2,44	0,583	8,096	4,49	10	196
ADCRECV	B	XC3S200-4FT256	21	1920	0	0	0	0	27	173	45	4,865	4,55	0,865	9,572	2,94	3	159
ADCRECV	B	XC3S200-4PQ208	21	1920	0	0	0	0	27	141	45	4,926	3,83	1,137	9,699	2,94	3	159
ADCRECV	B	XC3S200-4TQ144	21	1920	0	0	0	0	27	97	45	5,289	2,78	1,147	9,918	2,91	2	160
ADCRECV	B	XC3S200-4VQ100	21	1920	0	0	0	0	27	63	45	7,076	4,72	0,479	9,936	2,92	2	159
ADCRECV	B	XC3S200-5FT256	21	1920	0	0	0	0	27	173	45	3,557	3,97	0,512	9,077	2,94	3	160
ADCRECV	B	XC3S200-5PQ208	21	1920	0	0	0	0	27	141	45	5,254	2,77	0,989	9,558	2,92	3	159
ADCRECV	B	XC3S200-5TQ144	21	1920	0	0	0	0	27	97	45	5,461	2,93	0,627	8,941	2,94	2	160
ADCRECV	B	XC3S200-5VQ100	21	1920	0	0	0	0	27	63	45	5,516	3,16	1,016	9,479	2,91	2	160
ADCRECV	B	XC3S200A-4FG320	21	1792	0	0	0	0	27	248	27	4,757	3,83	0,912	9,781	3,57	6	180
ADCRECV	B	XC3S200A-4FT256	21	1792	0	0	0	0	27	195	27	5,601	4,01	0,545	9,974	3,56	6	180
ADCRECV	B	XC3S200A-4VQ100	21	1792	0	0	0	0	27	68	27	4,407	4,27	0,712	9,549	3,56	5	179
ADCRECV	B	XC3S200A-5FG320	21	1792	0	0	0	0	27	248	27	3,003	2,6	0,891	8,262	3,54	6	179
ADCRECV	B	XC3S200A-5FT256	21	1792	0	0	0	0	27	195	27	4,806	2,97	0,902	9,842	3,57	6	179
ADCRECV	B	XC3S200A-5VQ100	21	1792	0	0	0	0	27	68	27	3,664	3,13	0,501	8,719	3,56	6	179
ADCRECV	B	XC3S200AN-4FTG256	21	1792	0	0	0	0	27	195	31	5,601	4,01	0,545	9,974	3,58	6	180
ADCRECV	B	XC3S200AN-5FTG256	21	1792	0	0	0	0	27	195	31	4,806	2,97	0,902	9,842	3,56	6	180
ADCRECV	B	XC3S250E-4CP132	21	2448	0	0	0	0	27	92	57	5,818	6,23	1,156	9,622	3,33	5	169
ADCRECV	B	XC3S250E-4FT256	21	2448	0	0	0	0	27	172	57	4,908	2,91	1,215	9,953	3,34	5	169
ADCRECV	B	XC3S250E-4PQ208	21	2448	0	0	0	0	27	158	57	5,408	3,86	0,993	9,261	3,31	5	170
ADCRECV	B	XC3S250E-4TQ144	21	2448	0	0	0	0	27	108	57	5,734	4,43	0,888	9,83	3,33	5	169
ADCRECV	B	XC3S250E-4VQ100	21	2448	0	0	0	0	27	66	57	5,372	2,47	0,579	9,964	3,33	5	169
ADCRECV	B	XC3S250E-5CP132	21	2448	0	0	0	0	27	92	57	4,377	4,9	1,188	8,468	3,41	5	169
ADCRECV	B	XC3S250E-5FT256	21	2448	0	0	0	0	27	172	57	4,179	2,74	1,247	9,123	3,33	5	170
ADCRECV	B	XC3S250E-5PQ208	21	2448	0	0	0	0	27	158	57	4,706	3,21	1,054	9,648	3,32	5	169
ADCRECV	B	XC3S250E-5TQ144	21	2448	0	0	0	0	27	108	57	4,989	3,7	0,962	9,366	3,32	5	169
ADCRECV	B	XC3S250E-5VQ100	21	2448	0	0	0	0	27	66	57	4,676	2,1	0,637	8,964	3,32	4	169

Circuito	Est.	Tecnologia	UL_u	UL_d	Bits_u	Bits_d	Mult_u	Mult_d	Pines_u	Pines_d	Potencia	Reg2Reg	Tsu	Th	Tco	T_sint	T_imp	Memo
ADCRECV	B	XC3S400-4FG320	21	3584	0	0	0	0	27	221	64	5,152	2,93	1,169	9,787	3,05	3	163
ADCRECV	B	XC3S400-4FG456	21	3584	0	0	0	0	27	264	64	3,897	2,67	1,192	9,8	3,07	4	163
ADCRECV	B	XC3S400-4FT256	21	3584	0	0	0	0	27	173	64	5,357	3,23	1,044	9,523	3,05	3	163
ADCRECV	B	XC3S400-4PQ208	21	3584	0	0	0	0	27	141	64	6,492	3,66	0,562	9,934	3,03	3	163
ADCRECV	B	XC3S400-4TQ144	21	3584	0	0	0	0	27	97	64	6,394	5,25	-0,3	9,864	3,01	3	162
ADCRECV	B	XC3S400-5FG320	21	3584	0	0	0	0	27	221	64	3,986	2,61	1,01	9,257	3,03	3	162
ADCRECV	B	XC3S400-5FG456	21	3584	0	0	0	0	27	264	64	3,367	2,71	1,037	8,588	3,09	4	163
ADCRECV	B	XC3S400-5FT256	21	3584	0	0	0	0	27	173	64	3,778	2,38	0,904	8,902	3,06	3	162
ADCRECV	B	XC3S400-5PQ208	21	3584	0	0	0	0	27	141	64	3,968	3,62	1	9,11	3,01	3	163
ADCRECV	B	XC3S400-5TQ144	21	3584	0	0	0	0	27	97	64	6,678	4,36	0,345	9,689	3,06	3	163
ADCRECV	B	XC3S400A-4FG320	21	3584	0	0	0	0	27	251	31	5,508	3,09	0,905	9,532	3,77	7	183
ADCRECV	B	XC3S400A-4FG400	21	3584	0	0	0	0	27	311	31	3,686	3,97	0,886	8,797	3,77	7	183
ADCRECV	B	XC3S400A-4FT256	21	3584	0	0	0	0	27	195	31	5,285	3,51	0,919	9,736	3,77	7	183
ADCRECV	B	XC3S400A-5FG320	21	3584	0	0	0	0	27	251	31	3,987	2,14	0,868	8,863	3,78	7	183
ADCRECV	B	XC3S400A-5FG400	21	3584	0	0	0	0	27	311	31	3,997	3,08	0,859	8,902	3,75	7	183
ADCRECV	B	XC3S400A-5FT256	21	3584	0	0	0	0	27	195	31	4,305	3,08	0,816	8,957	3,75	6	183
ADCRECV	B	XC3S400AN-4FGG400	21	3584	0	0	0	0	27	311	35	3,686	3,97	0,886	8,797	3,73	7	183
ADCRECV	B	XC3S400AN-5FGG400	21	3584	0	0	0	0	27	311	35	3,997	3,08	0,859	8,902	3,77	7	183
ADCRECV	B	XC3S500E-4CP132	21	4656	0	0	0	0	27	92	86	5,39	4,46	1,021	8,853	3,49	6	174
ADCRECV	B	XC3S500E-4FG320	21	4656	0	0	0	0	27	232	85	4,545	5,28	1,255	9,926	3,47	7	173
ADCRECV	B	XC3S500E-4FT256	21	4656	0	0	0	0	27	190	86	5,129	3,32	0,819	9,139	3,49	6	174
ADCRECV	B	XC3S500E-4PQ208	21	4656	0	0	0	0	27	158	86	5,388	3,6	1,464	9,807	3,49	6	173
ADCRECV	B	XC3S500E-4VQ100	21	4656	0	0	0	0	27	66	86	7,642	5,51	1,032	8,776	3,45	6	173
ADCRECV	B	XC3S500E-5CP132	21	4656	0	0	0	0	27	92	86	4,756	3,96	1,486	9,085	3,47	6	173
ADCRECV	B	XC3S500E-5FG320	21	4656	0	0	0	0	27	232	85	3,956	4,44	1,282	9,036	3,46	7	173
ADCRECV	B	XC3S500E-5FT256	21	4656	0	0	0	0	27	190	86	4,464	2,97	1,465	8,419	3,47	6	173
ADCRECV	B	XC3S500E-5PQ208	21	4656	0	0	0	0	27	158	86	4,622	2,71	1,253	9,49	3,48	6	173

Circuito	Est.	Tecnologia	UL_u	UL_d	Bits_u	Bits_d	Mult_u	Mult_d	Pines_u	Pines_d	Potencia	Reg2Reg	Tsu	Th	Tco	T_sint	T_imp	Memo
ADCRECV	B	XC3S50-4CP132	21	768	0	0	0	0	27	89	32	4,853	4,2	1,005	9,85	2,84	2	157
ADCRECV	B	XC3S50-4PQ208	21	768	0	0	0	0	27	124	32	3,77	3,87	0,662	8,856	2,84	2	157
ADCRECV	B	XC3S50-4TQ144	21	768	0	0	0	0	27	97	32	5,843	3,34	0,999	9,616	2,84	2	157
ADCRECV	B	XC3S50-4VQ100	21	768	0	0	0	0	27	63	32	5,973	3,22	0,742	9,645	2,86	2	157
ADCRECV	B	XC3S50-5CP132	21	768	0	0	0	0	27	89	32	4,743	2,92	0,642	9,893	2,86	2	157
ADCRECV	B	XC3S50-5PQ208	21	768	0	0	0	0	27	124	32	2,857	3,39	0,833	8,219	2,85	3	157
ADCRECV	B	XC3S50-5TQ144	21	768	0	0	0	0	27	97	32	4,885	3,01	0,601	8,692	2,84	2	157
ADCRECV	B	XC3S50-5VQ100	21	768	0	0	0	0	27	63	32	3,761	3,24	0,65	9,257	2,86	2	157
ADCRECV	B	XC3S50A-4FT256	21	704	0	0	0	0	27	144	15	3,704	3,15	0,3	9,094	3,41	4	176
ADCRECV	B	XC3S50A-4TQ144	21	704	0	0	0	0	27	108	15	3,33	4,01	0,33	8,877	3,41	4	176
ADCRECV	B	XC3S50A-4VQ100	21	704	0	0	0	0	27	68	15	4,487	3,44	0,295	9,467	3,42	4	175
ADCRECV	B	XC3S50A-5FT256	21	704	0	0	0	0	27	144	15	2,83	2,54	0,299	8,211	3,42	4	175
ADCRECV	B	XC3S50A-5TQ144	21	704	0	0	0	0	27	108	15	3,112	2,71	0,331	7,817	3,42	4	176
ADCRECV	B	XC3S50A-5VQ100	21	704	0	0	0	0	27	68	15	3,158	3,12	0,304	8,145	3,44	4	175
ADCRECV	B	XC3S50AN-4TQG144	21	704	0	0	0	0	27	108	18	3,33	4,01	0,33	8,877	3,43	4	175
ADCRECV	B	XC3S50AN-5TQG144	21	704	0	0	0	0	27	108	18	3,112	2,71	0,331	7,817	3,43	4	176
ADCRECV	B	XC3S700A-4FG400	21	5888	0	0	0	0	27	311	37	4,136	3,84	0,985	9,975	4,03	8	188
ADCRECV	B	XC3S700A-4FG484	21	5888	0	0	0	0	27	372	37	3,464	4,32	0,939	9,437	3,97	8	187
ADCRECV	B	XC3S700A-4FT256	21	5888	0	0	0	0	27	161	37	4,742	4,08	0,584	9,972	4,01	8	188
ADCRECV	B	XC3S700A-5FG400	21	5888	0	0	0	0	27	311	37	3,369	2,91	0,949	9,491	3,97	8	188
ADCRECV	B	XC3S700A-5FG484	21	5888	0	0	0	0	27	372	37	3,126	3,94	0,905	8,042	3,98	8	187
ADCRECV	B	XC3S700A-5FT256	21	5888	0	0	0	0	27	161	38	3,25	3,64	0,967	9,069	3,98	8	188
ADCRECV	B	XC3S700AN-4FGG484	21	5888	0	0	0	0	27	372	42	3,464	4,32	0,939	9,437	3,97	8	187
ADCRECV	B	XC3S700AN-5FGG484	21	5888	0	0	0	0	27	372	42	3,126	3,94	0,905	8,042	4,02	8	186

Tabla 68. Registros de la tabla DatosReports correspondientes al circuito denominado ADCRECV

Al haber utilizado Microsoft Access para generar la base de datos de las aplicaciones desarrolladas, se cuenta con la ventaja de su sencillo interfaz de cara a construir consultas que filtren información. Esta capacidad va a permitir extraer y agrupar datos para resolver algunos de los objetivos propuestos al inicio del documento como los que a continuación se detallan.

7.2.1.1 Comparativa entre estrategias Balanced-Tiempo

Aunque el sistema de puntuaciones está basado en los resultados obtenidos aplicando la estrategia de equilibrio, *Balanced*, uno de los objetivos de este trabajo era comprobar si su utilización ofrecía buenos resultados. Por esta razón, el PdC se ejecutó utilizando tanto esta estrategia como otra de las propuestas por los fabricantes denominada aqui *Tiempo.* La estrategia *Tiempo* persigue conseguir la mejor frecuencia de trabajo sacrificando otros parámetros, esto es, área ó tiempo de CPU.

Para cotejar ambas estrategias se plantea una tabla que enfrenta el tiempo máximo entre dos registros, *Reg2Reg,* extraído para cada tecnología y circuito, debido a su extensión solo una parte de ella se muestra en la Tabla 69.

Circuito	Tecnologia	Reg2Reg (B)	Reg2Reg (T)	Mejor tiempo	
ADCRECV	EP2C8F256C7	3,568	3,064	Tiempo	16,45%
ADCRECV	EP2C8F256C8	4,015	3,849	Tiempo	4,31%
ADCRECV	EP2C8F256I8	4,015	3,849	Tiempo	4,31%
ADCRECV	EP3C16F484C8	3,905	3,702	Tiempo	5,48%
ADCRECV	EP3C16M164C7	3,627	3,503	Tiempo	3,54%
ADCRECV	EP3C16M164C8	3,976	3,756	Tiempo	5,86%
ADCRECV	EP3C16Q240C8	4,157	3,657	Tiempo	13,67%
ADCRECV	EP3C16U256C7	3,808	3,569	Tiempo	6,70%

Circuito	Tecnologia	Reg2Reg (B)	Reg2Reg (T)	Mejor tiempo	
ADCRECV	EP3C16U256C8	4,284	3,779	Tiempo	13,36%
ADCRECV	XA3S100E-4CPG132	4,601	3,963	Tiempo	16,10%
ADCRECV	XA3S100E-4TQG144	3,826	4,528	Balanced	-15,50%
ADCRECV	XA3S100E-4VQG100	5,13	6,576	Balanced	-21,99%
ADCRECV	XA3S1200E-4FTG256	6,187	4,359	Tiempo	41,94%
ADCRECV	XA3S200A-4FTG256	5,601	4,518	Tiempo	23,97%
ADCRECV	XA3S250E-4CPG132	5,818	5,586	Tiempo	4,15%
ADCRECV	XA3S250E-4FTG256	4,908	3,976	Tiempo	23,44%
ADCRECV	XA3S250E-4PQG208	5,408	4,609	Tiempo	17,34%
ADCRECV	XA3S250E-4TQG144	5,734	4,206	Tiempo	36,33%
ADCRECV	XA3S250E-4VQG100	5,372	5,458	Balanced	-1,58%
ADCRECV	XA3S400A-4FGG400	3,686	4,106	Balanced	-10,23%
ADCRECV	XA3S400A-4FTG256	5,285	4,729	Tiempo	11,76%

Tabla 69. Confrontación de tiempos entre registros (reg2reg) para Balanced y Tiempo

Las cuatro primeras columnas recogen el nombre del circuito, tecnología y los tiempos entre registros para la estrategia *Balanced y Tiempo*, respectivamente. La quinta columna muestra el resultado de comprobar qué tiempo es menor, el de *Balanced* o el de *Tiempo.* Por último, la sexta columna evalúa el porcentaje en que el retardo entre registros usando *Tiempo* es mejor respecto al de *Balanced*. Los valores negativos se presentan en los casos contrarios a lo esperado: la estrategia *Balanced* ofrece un retardo entre registros menor que la combinación circuito-tecnología equivalente ante la estrategia *Tiempo.*

En primer lugar, la Tabla 69 se dividió en dos grupos atendiendo al fabricante para evaluar el comportamiento de cada herramienta con cada estrategia. En el caso de Quartus II de Altera el porcentaje de circuitos-tecnología que mejoran con la estrategia *Tiempo* es del 93,27%, en la herramienta ISE Webpack este valor desciende hasta el 81,14%, Tabla 70 *y* Tabla 71.

Estrategia	Nº casos
Balanced	73
Tiempo	997

Tabla 70. Número de casos con mejor tiempo entre registros por estrategia para QuartusII

Estrategia	Nº casos
Balanced	244
Tiempo	1050

Tabla 71. Número de casos con mejor tiempo entre registros por estrategia para ISE Webpack

En segundo lugar, se calcula por cada fabricante, el porcentaje de casos en que la estrategia Tiempo ofrece mejores resultados que la Balanced, agrupándolos según el porcentaje de mejora, Tabla 72 y Tabla 73.

Rango (%)	Nº casos	% casos
<0%:	73	6,82%
[0% -10 %):	403	37,66%
[10% -20 %):	318	29,72%
[20% -30 %):	207	19,35%
[30% -40 %):	55	5,14%
[40% -50 %):	9	0,84%
[50% -60 %):	3	0,28%
[60% -70 %):	0	0,00%
[70% -80 %):	0	0,00%
[80% -90 %):	0	0,00%
[90% -100 %):	0	0,00%
>=100 % :	2	0,19%
TOTAL:	**1070**	**100,00%**

Tabla 72. Estudio de porcentajes en Quartus II

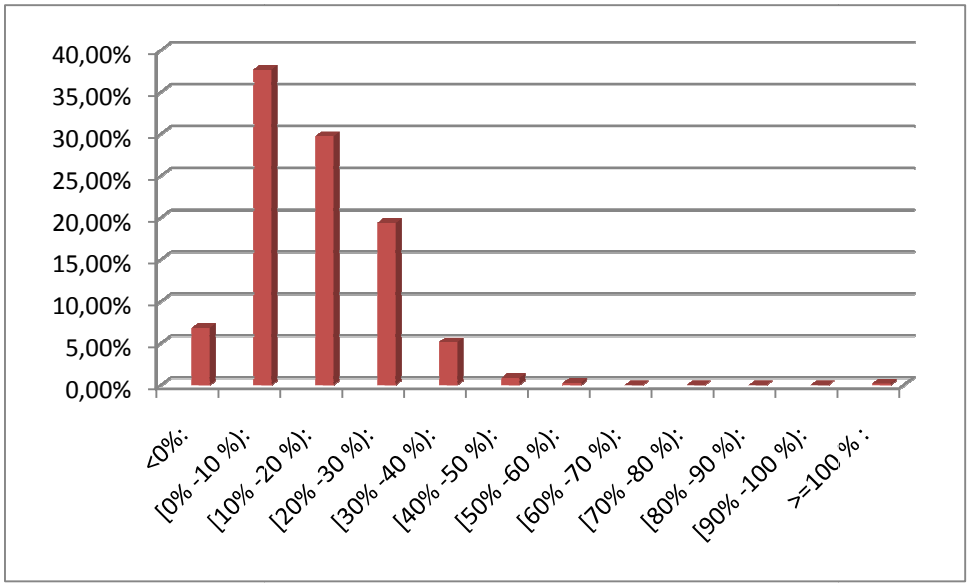


Figura 64. Gráfico de frecuencia de porcentajes de mejora Tiempo-Balanced (QuartusII)

Rango (%)	Nº casos	% casos
<0%:	244	18,86%
[0% -10 %):	510	39,41%
[10% -20 %):	342	26,43%
[20% -30 %):	136	10,51%
[30% -40 %):	32	2,47%
[40% -50 %):	18	1,39%
[50% -60 %):	5	0,39%
[60% -70 %):	1	0,08%
[70% -80 %):	3	0,23%
[80% -90 %):	2	0,15%
[90% -100 %):	1	0,08%
>=100 % :	0	0,00%
TOTAL:	1294	18,86%

Tabla 73. Estudio de porcentajes en ISE Webpack

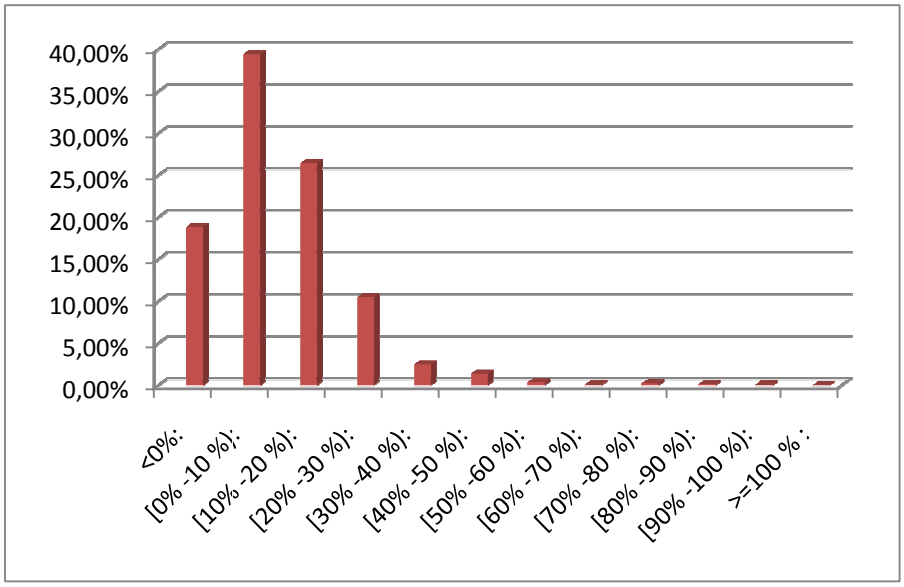


Figura 65. Gráfico de frecuencia de porcentajes de mejora Tiempo-Balanced (ISE Webpack)

Las dos tablas anteriores evidencian que las estrategias por defecto propuesta por los fabricantes consiguen un peor rendimiento que aquella cuyo objetivo es la velocidad. La estrategia *Tiempo* consigue reducir el retardo entre registros (Reg2Reg) en torno a un 10-20% para un 65% aproximadamente de los casos en ambas herramientas.

Los números obtenidos recuerdan los objetivos principales de ambas estrategias, y la necesidad de optar por una que persiga mejorar el tiempo además de incorporar restricciones al diseño si se pretende conseguir una frecuencia alta de trabajo. Sin embargo, si no es el rendimiento el objetivo principal, se concluye que la herramienta ISE Webpack ofrece resultados aceptables en su configuración por defecto. La mejora que supone la estrategia *tiempo* es superior al 20% en solo un 15% de los casos en ISE y de un 25% en QuartusII, por tanto los valores obtenidos con *Balanced* no son demasiado malos considerando que consigue mejores resultados en ocupación lógica y en tiempo de CPU, en especial en ISEWebPack.

7.2.1.2 Equivalencia entre los recursos lógicos de Xilinx y Altera

Aunque no forma parte de los objetivos iniciales de la tesis, el cuantioso conjunto de datos obtenido tras la ejecución del PdC proporciona un buen punto de partida para tratar de establecer una relación numérica entre las unidades lógicas que integran los dos grupos tecnológicos tratados.

Establecer una equivalencia entre los *LE* de Altera y los *LC*[32] de Xilinx, ha sido una labor perseguida durante décadas y nunca resuelta satisfactoriamente. En sus hojas de características, Xilinx destaca que gracias al resto de circuitería presente en sus *slices*, se cumple que

1 Slice = 2.25 LC

Por otra parte, Altera ha realizado algunas pruebas comparativas llegando a la conclusión de que

1 LE = 1.125 LC

En esta ocasión, se tratará de establecer una equivalencia en base a *slices* y LEs ya que es la información presente en los informes generados al término del PdC.

En algunos de los circuitos del PdC, las herramientas han hecho uso de otros recursos lógicos diferentes como los multiplicadores o los bloques de memoria. Ya que cada herramienta organiza y utiliza los recursos de sus tecnologías siguiendo diferentes caminos, se prescinde de aquellos circuitos

[32] Logic Cell es como denomina Xilinx a la pareja formada por una LUT y un elemento de almacenamiento. Cada slice incluye dos.

que en su construcción hacen uso de elementos que no sean unidades lógicas.

Esta consideración reduce de 16 a 11 los circuitos participantes en la comparativa. A continuación se eliminan aquellos que no tuvieron éxito sobre un número grande de tecnologías, quedando finalmente el total reducido a 7 circuitos.

Nuevamente, se cuenta con la tabla *DatosReports,* Tabla 68, para extraer la información relativa a las unidades lógicas utilizadas para implementar esos 7 circuitos, en concreto se lee el campo UL_u para la estrategia *Balanced*. Se obtiene una tabla por fabricante. En la Tabla 74 se muestra una parte de la original, demasiado extensa para incluirse en este documento.

Circuito	Tecnologia	UL_u
DDS_SYNTHESIZER	EP2C15AF256C6	101
DDS_SYNTHESIZER	EP2C15AF256C7	101
DDS_SYNTHESIZER	EP2C15AF256C8	101
DDS_SYNTHESIZER	EP2C15AF256I8	101
DDS_SYNTHESIZER	EP3C5U256C7	101
DDS_SYNTHESIZER	EP3C5U256C8	101
DDS_SYNTHESIZER	EP3C5U256I7	101
	Promedio	101,00
	DesvEst	0,00
DDS_SYNTHESIZER	XA3S100E-4CPG132	57
DDS_SYNTHESIZER	XA3S100E-4TQG144	57
DDS_SYNTHESIZER	XA3S100E-4VQG100	57
DDS_SYNTHESIZER	XC3S700A-5FT256	55
DDS_SYNTHESIZER	XC3S700AN-4FGG484	55
DDS_SYNTHESIZER	XC3S700AN-5FGG484	55
	Promedio	55,63
	DesvEst	0,87

Tabla 74. Detalle de algunos valores de 1 de los 7 circuitos utilizados en la comparativa de unidades lógicas

Una de las características que llama principalmente la atención en esta tabla, quizás no en la porción mostrada, es que especialmente en las tecnologías de Xilinx, un mismo circuito utiliza cantidades diferentes de unidades lógicas en función del modelo que utilice como objetivo.

Esta razón obliga a calcular el promedio de unidades lógicas empleadas por fabricante y circuito y establecer la relación entre los recursos de ambas marcas a partir de este valor. En la Tabla 75 se recoge el cálculo efectuado en cada caso junto a la desviación estándar de cada muestra.

Nombre_circuito	BALANCED				RELACION (U.L. Altera / U.L. Xilinx)
	Altera		Xilinx		
	LE		Slices		
	Promedio	Desvest	Promedio	Desvest	
ADCRECV	**38,00**	**0,00**	**21,00**	**0,00**	**1,81**
DDS_SYNTHESIZER	101,00	0,00	55,63	0,87	1,82
I2CSLAVETOP	221,84	0,37	168,23	6,11	1,32
SC_CORPROC	1056,59	0,49	384,94	9,50	2,74
SIN	**40,00**	**0,00**	**16,00**	**0,00**	**2,50**
SPI_BOOT	104,00	0,00	66,19	1,67	1,57
UCRC_PAR	**24,00**	**0,00**	**11,00**	**0,00**	**2,18**
	PROMEDIO:				**1,99**

Rango	Frecuencia
>= 2,5	2
(2-2,5)	1
(1,5-2)	3
(1-1,5)	1
TOTAL:	7

Tabla 75. Promedio y desviación estándar de los siete circuitos. Relación entre las UL de cada fabricante

La Tabla 75 refleja para los circuitos incluidos en la comparativa que la desviación estándar de la muestra no es

demasiado alta en ninguno de los casos, lo que permite trabajar con los promedios sin cometer un error demasiado importante. Por otro lado, vuelve a manifestarse que es en las tecnologías de Xilinx donde se observa mayor diferencia entre la cantidad de U.L. empleadas para implementar un mismo circuito en diferentes modelos de FPGAs.

A partir de la cantidad de unidades lógicas promedio se establece la relación entre fabricantes. De forma inmediata se comprueba que la razón no es constante sino que varía con cada circuito. La conclusión a la que se llega es que no es posible establecer una equivalencia directa entre la lógica de ambas arquitecturas, ni siquiera si usamos exclusivamente los circuitos con desviación estándar cero, destacados en negrita en la Tabla 75.

De todos modos, es interesante ver que calculando el promedio de todas las relaciones obtenidas, el valor resultante coincide con la suposición teórica que a priori se plantea con tan solo echar un vistazo a las hojas de características de cada arquitectura: “Si 1 *slice* de Xilinx consta de dos LUTs de 4 entradas y dos elementos de almacenamiento, y el *LE* de Altera se construye con una LUT de 4 entradas y un solo elemento de almacenamiento, un slice equivale a dos LE”.

Hay que decir a favor de Xilinx, que su herramienta ISE informa del total de *slices* que se han utilizado, independientemente de que la ocupación de éste sea completa o no, es decir, use sus dos LCs o no, condición que favorece a Altera en la relación.

7.2.2. Fallos detectados en la ejecución: Tabla *Fallos*

La base de datos incluye una tabla denominada *Fallos* encargada del almacenar posibles anomalías acontecidas durante la ejecución del PdC. Una consulta de la misma permite conocer qué circuito-tecnología falló en algunas de las fases. Cada registro de la tabla incluye información sobre circuito, tecnología, estrategia y ruta donde debía encontrarse el informe no hallado.

La aplicación **ResultadosPdC** muestra esta tabla con un total de 20 registros, Tabla 76. Los fallos detectados son debido a dos causas principales:

- Tecnologías de Altera, 20 registros: no son propiamente fallos de ejecución, realmente son problemas durante la implementación. Se trata de 10 variantes de los modelos de Cyclone3, EP3C5E144, EP3C10E144 y EP3C16M164, frente a las dos estrategias, que aunque cuentan con 94 terminales de E/S de usuario, solo pueden configurar 88 en el estándar de salida 2,5V LVTTL/LVCMOS. Esta cantidad es insuficiente para el circuito T400_CORE que necesita de 89, y los informes no lo muestran como un intento de utilización de pines superior al 100%.

Tecnología	Circuito	Estrategia	F_imp	F_Potencia	F_tiempo
EP3C10E144A7	T400_CORE	BALANCED		C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C10E144A7	C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C10E144A7
EP3C10E144C7	T400_CORE	BALANCED		C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C10E144C7	C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C10E144C7
EP3C10E144C8	T400_CORE	BALANCED		C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C10E144C8	C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C10E144C8
EP3C10E144I7	T400_CORE	BALANCED		C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C10E144I7	C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C10E144I7
EP3C16M164C7	T400_CORE	BALANCED		C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C16M164C7	C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C16M164C7
EP3C16M164C8	T400_CORE	BALANCED		C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C16M164C8	C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C16M164C8
EP3C5E144A7	T400_CORE	BALANCED		C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C5E144A7	C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C5E144A7
EP3C5E144C7	T400_CORE	BALANCED		C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C5E144C7	C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C5E144C7
EP3C5E144C8	T400_CORE	BALANCED		C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C5E144C8	C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C5E144C8
EP3C5E144I7	T400_CORE	BALANCED		C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C5E144I7	C:\RESULTADOS_ALTERA\Automocion_ALTERA_BALANCED\t400_core\EP3C5E144I7
EP3C10E144A7	T400_CORE	TIEMPO		C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C10E144A7	C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C10E144A7
EP3C10E144C7	T400_CORE	TIEMPO		C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C10E144C7	C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C10E144C7
EP3C10E144C8	T400_CORE	TIEMPO		C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C10E144C8	C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C10E144C8

Tecnología	Circuito	Estrategia	F_imp	F_Potencia	F_tiempo
EP3C10E144I7	T400_CORE	TIEMPO		C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C10E144I7	C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C10E144I7
EP3C16M164C7	T400_CORE	TIEMPO		C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C16M164C7	C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C16M164C7
EP3C16M164C8	T400_CORE	TIEMPO		C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C16M164C8	C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C16M164C8
EP3C5E144A7	T400_CORE	TIEMPO		C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C5E144A7	C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C5E144A7
EP3C5E144C7	T400_CORE	TIEMPO		C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C5E144C7	C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C5E144C7
EP3C5E144C8	T400_CORE	TIEMPO		C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C5E144C8	C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C5E144C8
EP3C5E144I7	T400_CORE	TIEMPO		C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C5E144I7	C:\RESULTADOS_ALTERA\Automocion_ALTERA_TIEMPO\t400_core\EP3C5E144I7

Tabla 76. Tabla *Fallos* de la base de datos. 20 registros.

7.2.3. Fallos de implementación: Tabla *Sobredimensionados*

La aplicación **ResultadosPdC** ha detectado un total de 1.157 fallos de implementación, ocasionados éstos por una sobreutilización de recursos lógicos o de pines de usuario. El desglose de fallos se detalla en la Tabla 77.

	Xilinx (Balanced)	Xilinx (Tiempo)	Altera (Balanced)	Altera (Tiempo)
Total Pines y lógica	126	149	36	40
Total pines	124	101	0	11
Total lógica	149	141	139	135
TOTAL REGISTROS:	399	399	175	186

Tabla 77. Desglose de fallos de implementación

A la vista de los resultados registrados en la tabla *Sobredimensionados* se aprecia que el circuito incluido en el PdC denominado AES128_FAST, es demasiado grande para la muestra de FPGAs utilizadas como objetivo. Casi todas las tecnologías carecen de recursos lógicos y pines de usuario suficientes para recibir este diseño. Otros circuitos como RS_DEC o RS_ENCODE necesitan un gran número de unidades lógicas y de terminales de usuarios, respectivamente.

El resto de registros recogen otras combinaciones tecnología-circuito con desenlaces previsibles como no poder ser implementados en FPGAs con escasa capacidad lógica.

En la Tabla 78, se presenta una muestra de algunos de los 1.151 registros que conforman esta tabla.

Fabricante	Circuito	Estrategia	Tecnologia	UL_u	UL_d	Pines_u	Pines_d
XILINX	AES_DEC	BALANCED	XC3S50-4CP132	915	768	31	89
XILINX	AES_DEC	BALANCED	XC3S50-4PQ208	915	768	31	124
XILINX	AES_DEC	BALANCED	XC3S50-4TQ144	915	768	31	97
XILINX	AES_DEC	BALANCED	XC3S50-4VQ100	915	768	31	63
XILINX	AES_DEC	BALANCED	XC3S50-5CP132	913	768	31	89
XILINX	AES_DEC	BALANCED	XC3S50-5PQ208	913	768	31	124
XILINX	AES_DEC	BALANCED	XC3S50-5TQ144	913	768	31	97
XILINX	AES_DEC	BALANCED	XC3S50-5VQ100	913	768	31	63
XILINX	AES_DEC	BALANCED	XC3S50A-4FT256	1007	704	31	144
XILINX	AES_DEC	BALANCED	XC3S50A-4TQ144	1007	704	31	108
XILINX	AES_DEC	BALANCED	XC3S50A-4VQ100	1007	704	31	68
ALTERA	AES128_FAST	BALANCED	EP2C5F256I8	9600	4608	262	158
ALTERA	AES128_FAST	BALANCED	EP2C5Q208C7	9600	4608	262	142
ALTERA	AES128_FAST	BALANCED	EP2C5Q208C8	9600	4608	262	142
ALTERA	AES128_FAST	BALANCED	EP2C5Q208I8	9600	4608	262	142
ALTERA	AES128_FAST	BALANCED	EP2C5T144C6	9600	4608	262	89
ALTERA	AES128_FAST	BALANCED	EP2C5T144C7	9600	4608	262	89
ALTERA	AES128_FAST	BALANCED	EP2C5T144C8	9600	4608	262	89
ALTERA	AES128_FAST	BALANCED	EP2C5T144I8	9600	4608	262	89

Tabla 78. Muestra de la tabla *Sobredimensionados* de la base de datos

7.3. Cálculo de las magnitudes y de Pk

Concluidas las fases de ejecución del PdC y de la lectura de datos, la aplicación **ResultadosPdC** procede al cálculo de las magnitudes necesarias para determinar las puntuaciones de cada tecnología. Las magnitudes que aparecen recogidas en la tabla *Puntuaciones,* de la que se presenta solo una parte debido a su extensión, cuenta con 6.139 registros, Tabla 79.

El incremento de registros en esta tabla respecto a las anteriores es notorio, la causa es que se agregan variantes que cubren diferentes rangos de temperatura de algunos de los modelos de Xilinx, lo que supone un aumento de 186 a 246 tecnologías en total. Las FPGAs son añadidas automáticamente por la aplicación en una primera fase que duplica los resultados obtenidos para Xilinx, uno para temperatura industrial y otro para la comercial. En una fase posterior algunos de estos nuevos modelos son finalmente excluidos al no ajustarse al objetivo del precio marcado, máximo 50 $ por unidad.

La tabla *Puntuaciones* recoge toda la información necesaria para calcular Pk y en base a ella llegar hasta la clasificación final de FPGAs. Pero además, incluye otra serie de datos interesantes como el tiempo de CPU o la memoria de pico empleados por cada herramienta en concluir hasta la fase de implementación.

A partir de las magnitudes disponibles en esta tabla, es posible establecer otras clasificaciones clásicas independientes del precio, por ejemplo, ordenar por cada circuito que modelo ofrece mejor frecuencia, Tabla 80, o bien según tiempo de CPU consumido, etc.

En la Tabla 80, se comprueba que para el circuito SIN, las 10 FPGAs que consiguen una mayor frecuencia de trabajo teórica,

pertenecen al fabricante Altera. Hay que avanzar hasta la posición vigésimo quinta para encontrar una de Xilinx.

Esta misma tabla almacena las puntuaciones Pk calculadas por **ResultadosPdC** a partir de la relación entre cada magnitud y el precio de la tecnología valorada.

Circuito	Est.	Tecnologia	F_MHz	UL_ sobra	Bits_ sobra	Mult18x18 _sobra	Pot_mW	Precio_$	Tiempo	Memo	P1	P2	P3	P4	P5
RS_DEC	B	EP2C15AF256C6	126,10	66,83	87,97	100,00	79	38,9	59	272	3,2417	1,7181	2,2615	2,2615	0,3254
FPU	B	EP2C15AF256C6	79,18	70,77	100,00	92,31	92	38,9	57	262	2,0354	1,8193	2,5707	2,5707	0,2794
AES_DEC	B	EP2C15AF256C6	102,86	84,40	98,29	100,00	76	38,9	36	256	2,6442	2,1696	2,5268	2,5268	0,3382
AES_ENC	B	EP2C15AF256C6	138,18	86,53	98,29	100,00	75	38,9	51	256	3,5522	2,2244	2,5268	2,5268	0,3428
SC_CORPROC	B	EP2C15AF256C6	210,66	92,69	100,00	100,00	78	38,9	11	253	5,4154	2,3828	2,5707	2,5707	0,3296
T400_CORE	B	EP2C15AF256C6	131,54	95,54	100,00	100,00	85	38,9	16	252	3,3816	2,4559	2,5707	2,5707	0,3024
UART_16750	B	EP2C15AF256C6	171,23	96,50	99,49	100,00	74	38,9	11	252	4,4019	2,4807	2,5576	2,5576	0,3474
ANALYTIC_FI LTER_H_A1	B	EP2C15AF256C6	318,88	98,41	99,96	100,00	77	38,9	9	251	8,1974	2,5298	2,5698	2,5698	0,3339
I2CSLAVETOP	B	EP2C15AF256C6	258,60	98,47	100,00	100,00	70	38,9	8	250	6,6478	2,5314	2,5707	2,5707	0,3672
SPI_BOOT	B	EP2C15AF256C6	214,92	99,28	100,00	100,00	71	38,9	8	250	5,5248	2,5522	2,5707	2,5707	0,3621
DDS_SYNTHES IZER	B	EP2C15AF256C6	219,20	99,30	100,00	100,00	76	38,9	7	250	5,6350	2,5527	2,5707	2,5707	0,3382
SIN	B	EP2C15AF256C6	178,44	99,72	100,00	100,00	75	38,9	5	249	4,5872	2,5636	2,5707	2,5707	0,3428
ADCRECV	B	EP2C15AF256C6	363,11	99,74	100,00	100,00	73	38,9	5	249	9,3344	2,5639	2,5707	2,5707	0,3521
UCRC_PAR	B	EP2C15AF256C6	496,03	99,83	100,00	100,00	71	38,9	5	249	12,751 5	2,5664	2,5707	2,5707	0,3621
FPU	B	XC3S700AN-5FGG484C	22,06	7,52	100,00	80,00	52	44,58	498,89	401	0,4949	0,1688	2,2432	1,7945	0,4314
RS_DEC	B	XC3S700AN-5FGG484C	113,31	42,09	45,00	100,00	50	44,58	114,17	276	2,5418	0,9440	1,0094	2,2432	0,4486
AES_ENC	B	XC3S700AN-5FGG484C	141,26	87,67	85,00	100,00	44	44,58	53,14	206	3,1688	1,9666	1,9067	2,2432	0,5098
RS_ENCODE	B	XC3S700AN-5FGG484C	153,12	88,59	100,00	100,00	48	44,58	30,96	193	3,4346	1,9871	2,2432	2,2432	0,4673

Circuito	Est.	Tecnologia	F_MHz	UL_ sobra	Bits_ sobra	Mult18x18 _sobra	Pot_mW	Precio_$	Tiempo	Memo	P1	P2	P3	P4	P5
AES_DEC	B	XC3S700AN-5FGG484C	112,49	88,74	85,00	100,00	44	44,58	50,82	202	2,5232	1,9906	1,9067	2,2432	0,5098
T400_CORE	B	XC3S700AN-5FGG484C	89,53	93,14	100,00	100,00	47	44,58	33,68	200	2,0082	2,0892	2,2432	2,2432	0,4773
SC_CORPROC	B	XC3S700AN-5FGG484C	179,44	93,31	100,00	100,00	42	44,58	28,48	193	4,0250	2,0931	2,2432	2,2432	0,5341
UART_16750	B	XC3S700AN-5FGG484C	125,50	93,87	95,00	100,00	42	44,58	27,17	193	2,8152	2,1056	2,1310	2,2432	0,5341
I2CSLAVETOP	B	XC3S700AN-5FGG484C	164,74	97,15	100,00	100,00	40	44,58	19,25	189	3,6955	2,1792	2,2432	2,2432	0,5608
SPI_BOOT	B	XC3S700AN-5FGG484C	155,33	98,83	100,00	100,00	40	44,58	15,49	188	3,4842	2,2169	2,2432	2,2432	0,5608
DDS_SYNTHESIZER	B	XC3S700AN-5FGG484C	182,82	99,07	100,00	100,00	41	44,58	14,32	188	4,1008	2,2222	2,2432	2,2432	0,5471
ADCRECV	B	XC3S700AN-5FGG484C	319,90	99,64	100,00	100,00	42	44,58	12,02	186	7,1758	2,2352	2,2432	2,2432	0,5341
SIN	B	XC3S700AN-5FGG484C	157,28	99,73	100,00	100,00	38	44,58	11,99	188	3,5281	2,2371	2,2432	2,2432	0,5903
UCRC_PAR	B	XC3S700AN-5FGG484C	277,32	99,81	100,00	100,00	38	44,58	12,5	187	6,2206	2,2390	2,2432	2,2432	0,5903

Tabla 79. Algunos registros de la tabla Puntuaciones

Tecnología	Circuito	F_MHz
EP3C16F256C6	SIN	200,48
EP3C16U256C6	SIN	200,48
EP3C5F256C6	SIN	197,75
EP3C5U256C6	SIN	197,75
EP3C10F256C6	SIN	196,58
EP3C10U256C6	SIN	196,58
EP3C16F484C6	SIN	188,61
EP2C8T144C6	SIN	184,30
EP2C8F256C6	SIN	183,65
EP2C5T144C6	SIN	182,42
EP3C16F256C7	SIN	179,92
EP3C16U256C7	SIN	179,92
EP2C15AF256C6	SIN	178,44
EP2C5F256C6	SIN	177,34
EP3C16M164C7	SIN	171,38
EP3C5M164C7	SIN	171,23
EP3C5E144C7	SIN	170,74
EP3C16F484C7	SIN	170,62
EP3C16U484C7	SIN	170,62
EP3C16E144C7	SIN	168,61
EP3C5F256I7	SIN	167,25
EP3C5U256I7	SIN	167,25
EP3C5F256C7	SIN	165,70
EP3C5U256C7	SIN	165,70
XC3S250E-5PQ208C	SIN	164,04
EP3C5E144A7	SIN	163,80
XC3S400A-5FT256C	SIN	163,69
EP3C10F256A7	SIN	162,87
EP3C10F256C7	SIN	162,21
EP3C10U256C7	SIN	162,21
EP3C10F256I7	SIN	160,80
EP3C10U256I7	SIN	160,80
EP3C10E144C7	SIN	160,67
XC3S1200E-5FT256C	SIN	160,28
XC3S500E-5CP132C	SIN	160,28
XC3S500E-5FT256C	SIN	160,00
EP3C10E144I7	SIN	159,03
XC3S100E-5TQ144C	SIN	158,88
XC3S1400A-5FT256C	SIN	158,60
EP2C8Q208C7	SIN	158,53
XC3S250E-5VQ100C	SIN	157,70
EP2C8F256C7	SIN	157,55

Tecnología	Circuito	F_MHz
XC3S50A-5FT256C	SIN	157,55
XC3S50A-5TQ144C	SIN	157,55
XC3S50AN-5TQG144C	SIN	157,55
XC3S250E-5TQ144C	SIN	157,28
XC3S700A-5FG484C	SIN	157,28
XC3S700AN-5FGG484C	SIN	157,28
EP2C8T144C7	SIN	156,99
XC3S400A-5FG320C	SIN	156,96
XC3S500E-5FG320C	SIN	156,84
XC3S250E-5CP132C	SIN	156,54
EP3C16Q240C8	SIN	156,37
EP2C15AF484C7	SIN	155,74
EP3C10E144A7	SIN	155,21
XC3S500E-5PQ208C	SIN	155,11
XC3S200A-5VQ100C	SIN	154,80
XC3S400-5FG320C	SIN	154,54
XC3S50A-5VQ100C	SIN	153,68
EP2C15AF256C7	SIN	153,44
EP2C5T144C7	SIN	153,14
XC3S200A-5FT256C	SIN	152,74
XC3S200AN-5FTG256C	SIN	152,74
XC3S700A-5FT256C	SIN	152,70
EP3C5E144I7	SIN	152,23
EP3C25E144C8	SIN	151,29
XC3S100E-5CP132C	SIN	151,15
EP2C5F256C7	SIN	151,08
EP3C16F484C8	SIN	150,99
EP3C16U484C8	SIN	150,99
XC3S250E-5FT256C	SIN	150,53
EP3C25F256C8	SIN	150,40
EP3C25U256C8	SIN	150,40
EP2C5Q208C7	SIN	150,38
XC3S100E-5VQ100C	SIN	150,33
XC3S1000-5FT256I	SIN	150,04
XC3S1000-5FT256C	SIN	150,04
EP3C16E144C8	SIN	149,95
EP3C16F256C8	SIN	149,70
EP3C16U256C8	SIN	149,70
EP3C25F324C8	SIN	149,59
EP3C25Q240C8	SIN	149,25
EP3C16M164C8	SIN	149,08
EP3C10E144C8	SIN	149,05

Tecnología	Circuito	F_MHz
XC3S200-5VQ100C	SIN	148,99
EP3C10F256C8	SIN	148,50
EP3C10U256C8	SIN	148,50
EP3C5F256C8	SIN	148,46
EP3C5U256C8	SIN	148,46
XC3S400A-5FG400C	SIN	145,82
XC3S400AN-5FGG400C	SIN	145,82
XC3S700A-5FG400C	SIN	145,73
XC3S200A-5FG320C	SIN	145,45
XA3S250E-4PQG208Q	SIN	142,82
XA3S250E-4PQG208I	SIN	142,82
XC3S250E-4PQ208I	SIN	142,82
XC3S250E-4PQ208C	SIN	142,82
XC3S1200E-4FG320C	SIN	142,78
EP3C5E144C8	SIN	142,45
XA3S100E-4TQG144Q	SIN	141,00
XA3S100E-4TQG144I	SIN	141,00
XC3S100E-4TQ144I	SIN	141,00
XC3S100E-4TQ144C	SIN	141,00
XC3S50-5TQ144C	SIN	139,45
XA3S1200E-4FTG256I	SIN	139,43
XA3S500E-4CPG132Q	SIN	139,43
XA3S500E-4CPG132I	SIN	139,43
XC3S1200E-4FT256I	SIN	139,43
XC3S1200E-4FT256C	SIN	139,43
XC3S500E-4CP132I	SIN	139,43
XC3S500E-4CP132C	SIN	139,43
XA3S500E-4FTG256Q	SIN	139,30
XA3S500E-4FTG256I	SIN	139,30
XC3S500E-4FT256I	SIN	139,30
XC3S500E-4FT256C	SIN	139,30
EP2C20F484C8	SIN	137,61
XA3S250E-4VQG100Q	SIN	137,29
XA3S250E-4VQG100I	SIN	137,29
XC3S250E-4VQ100I	SIN	137,29
XC3S250E-4VQ100C	SIN	137,29
EP2C20F256C8	SIN	136,99
XA3S250E-4TQG144Q	SIN	136,91
XA3S250E-4TQG144I	SIN	136,91
XC3S250E-4TQ144I	SIN	136,91
XC3S250E-4TQ144C	SIN	136,91
XC3S500E-4FG320I	SIN	136,54

Tecnología	Circuito	F_MHz
XC3S500E-4FG320C	SIN	136,54
XA3S250E-4CPG132Q	SIN	136,28
XA3S250E-4CPG132I	SIN	136,28
XC3S250E-4CP132I	SIN	136,28
XC3S250E-4CP132C	SIN	136,28
EP2C8T144C8	SIN	136,00
EP2C8T144I8	SIN	136,00
XC3S700A-4FG400I	SIN	136,00
XC3S700A-4FG400C	SIN	136,00
EP2C20Q240C8	SIN	135,26
XA3S500E-4PQG208Q	SIN	134,99
XA3S500E-4PQG208I	SIN	134,99
XC3S500E-4PQ208I	SIN	134,99
XC3S500E-4PQ208C	SIN	134,99
XA3S400A-4FTG256Q	SIN	134,84
XA3S400A-4FTG256I	SIN	134,84
XC3S400A-4FT256I	SIN	134,84
XC3S400A-4FT256C	SIN	134,84
XC3S1400A-4FT256I	SIN	134,70
XC3S1400A-4FT256C	SIN	134,70
XC3S400-5FT256C	SIN	134,59
XC3S400-4FG320I	SIN	134,54
XC3S400-4FG320C	SIN	134,54
XA3S400A-4FGG400I	SIN	133,85
XC3S400A-4FG400I	SIN	133,85
XC3S400A-4FG400C	SIN	133,85
XC3S400AN-4FGG400I	SIN	133,85
XC3S400AN-4FGG400C	SIN	133,85
XC3S200A-4VQ100I	SIN	133,67
XC3S200A-4VQ100C	SIN	133,67
XC3S50A-4VQ100I	SIN	133,39
XC3S50A-4VQ100C	SIN	133,39
XC3S400-5FG456C	SIN	132,86
XC3S400A-4FG320I	SIN	132,42
XC3S400A-4FG320C	SIN	132,42
EP2C15AF484C8	SIN	132,26
EP2C15AF484I8	SIN	132,26
XA3S100E-4CPG132Q	SIN	131,60
XA3S100E-4CPG132I	SIN	131,60
XC3S100E-4CP132I	SIN	131,60
XC3S100E-4CP132C	SIN	131,60
XA3S250E-4FTG256Q	SIN	131,06

Tecnología	Circuito	F_MHz
XA3S250E-4FTG256I	SIN	131,06
XC3S250E-4FT256I	SIN	131,06
XC3S250E-4FT256C	SIN	131,06
EP2C5F256C8	SIN	131,01
EP2C5F256I8	SIN	131,01
XC3S700A-4FG484I	SIN	130,91
XC3S700A-4FG484C	SIN	130,91
XC3S700AN-4FGG484I	SIN	130,91
XC3S700AN-4FGG484C	SIN	130,91
XA3S100E-4VQG100Q	SIN	130,86
XA3S100E-4VQG100I	SIN	130,86
XC3S100E-4VQ100I	SIN	130,86
XC3S100E-4VQ100C	SIN	130,86
XC3S200A-4FG320I	SIN	130,65
XC3S200A-4FG320C	SIN	130,65
XC3S1000-4FT256I	SIN	130,53
XC3S1000-4FT256C	SIN	130,53
XC3S50-5PQ208C	SIN	130,40
XC3S50A-4FT256I	SIN	130,40
XC3S50A-4FT256C	SIN	130,40
XA3S200A-4FTG256Q	SIN	129,95
XA3S200A-4FTG256I	SIN	129,95
XC3S200A-4FT256I	SIN	129,95
XC3S200A-4FT256C	SIN	129,95
XC3S200AN-4FTG256I	SIN	129,95
XC3S200AN-4FTG256C	SIN	129,95
XC3S700A-4FT256I	SIN	129,92
XC3S700A-4FT256C	SIN	129,92
XC3S200-5TQ144C	SIN	129,77
XC3S200-4FT256I	SIN	129,74
XC3S200-4FT256C	SIN	129,74
XC3S200-4VQ100I	SIN	129,70
XC3S200-4VQ100C	SIN	129,70
EP2C8Q208C8	SIN	129,52
EP2C8Q208I8	SIN	129,52
XC3S200-5FT256C	SIN	129,45
XC3S50A-4TQ144I	SIN	129,45
XC3S50A-4TQ144C	SIN	129,45
XC3S50AN-4TQG144I	SIN	129,45
XC3S50AN-4TQG144C	SIN	129,45
XC3S400-5PQ208C	SIN	129,37
XC3S500E-4VQ100I	SIN	129,00

Tecnología	Circuito	F_MHz
XC3S500E-4VQ100C	SIN	129,00
EP2C8AF256I8	SIN	127,98
EP2C8F256C8	SIN	127,98
EP2C8F256I8	SIN	127,98
XC3S1000-4FG320I	SIN	127,88
XC3S1000-4FG320C	SIN	127,88
EP2C5Q208C8	SIN	127,65
EP2C5Q208I8	SIN	127,65
EP2C15AF256C8	SIN	127,50
EP2C15AF256I8	SIN	127,50
XC3S200-5PQ208C	SIN	126,02
XC3S50-4PQ208I	SIN	125,17
XC3S50-4PQ208C	SIN	125,17
XC3S50-5VQ100C	SIN	124,50
XC3S400-5TQ144C	SIN	124,21
XC3S50-5CP132C	SIN	122,73
XC3S50-4TQ144I	SIN	121,61
XC3S50-4TQ144C	SIN	121,61
XC3S400-4FG456I	SIN	119,89
XC3S400-4FG456C	SIN	119,89
XC3S200-4TQ144I	SIN	117,69
XC3S200-4TQ144C	SIN	117,69
XC3S400-4FT256I	SIN	114,76
XC3S400-4FT256C	SIN	114,76
XC3S400-4PQ208I	SIN	114,08
XC3S400-4PQ208C	SIN	114,08
EP2C5T144C8	SIN	110,82
EP2C5T144I8	SIN	110,82
XC3S200-4PQ208I	SIN	110,50
XC3S200-4PQ208C	SIN	110,50
XC3S50-4CP132I	SIN	109,02
XC3S50-4CP132C	SIN	109,02
XC3S400-4TQ144I	SIN	109,00
XC3S400-4TQ144C	SIN	109,00
XC3S50-4VQ100I	SIN	108,11
XC3S50-4VQ100C	SIN	108,11

Tabla 80. Clasificación de FPGAs atendiendo a la frecuencia conseguida para el circuito SIN

7.3.1. Fallos obtenidos en el cálculo de Pk

En el punto anterior se recordó como la aplicación duplicaba el número de registros de las tecnologías de Xilinx al construir la tabla *Puntuaciones.* Durante este proceso **ResultadosPdC** hace uso de la tabla *Características* con datos actualizados al mes de noviembre de 2009 y detecta si el modelo comercial e industrial contemplado para cada tecnología de Xilinx se ofrece a un precio inferior a los 50$.

En el caso de que un modelo no esté disponible en la tabla *Caracteristicas*, se considera un fallo del cálculo de puntuación Pk y se almacena en la tabla *Fallo_puntuaciones*. Lo mismo sucede en el caso de que algunas de las magnitudes hayan originado un valor no válido, esto es que la potencia o la frecuencia hayan resultado 0, situación que denota un error de las herramientas de análisis.

Fallo_puntuaciones devuelve un total de 1.244 registros ya que se repite el proceso por cada tecnología, circuito y estrategia. Para mayor comodidad se presentan solo los resultados correspondientes a la estrategia Balanced excluyendo las repeticiones. Se establecen varios grupos, fallos solo por precio, Tabla 81, y fallos solo por potencia 0mW,Tabla 82, y fallo por potencia y precio, Tabla 83. No se ha detectado ningún otro tipo de anomalía que afecte al cáculo de puntuaciones.

Tecnologia	Fallo_precio	Circuito
XA3S400A-4FGG400Q	VERDADERO	ADCRECV
XC3S100E-5CP132I	VERDADERO	ADCRECV
XC3S100E-5TQ144I	VERDADERO	ADCRECV
XC3S100E-5VQ100I	VERDADERO	ADCRECV
XC3S1200E-4FG320I	VERDADERO	ADCRECV
XC3S1200E-5FT256I	VERDADERO	ADCRECV
XC3S1400A-5FT256I	VERDADERO	ADCRECV

Tecnologia	Fallo_precio	Circuito
XC3S200-5FT256I	VERDADERO	ADCRECV
XC3S200-5PQ208I	VERDADERO	ADCRECV
XC3S200-5TQ144I	VERDADERO	ADCRECV
XC3S200-5VQ100I	VERDADERO	ADCRECV
XC3S200A-5FG320I	VERDADERO	ADCRECV
XC3S200A-5FT256I	VERDADERO	ADCRECV
XC3S200A-5VQ100I	VERDADERO	ADCRECV
XC3S200AN-5FTG256I	VERDADERO	ADCRECV
XC3S250E-5CP132I	VERDADERO	ADCRECV
XC3S250E-5FT256I	VERDADERO	ADCRECV
XC3S250E-5PQ208I	VERDADERO	ADCRECV
XC3S250E-5TQ144I	VERDADERO	ADCRECV
XC3S250E-5VQ100I	VERDADERO	ADCRECV
XC3S400-5FG320I	VERDADERO	ADCRECV
XC3S400-5FG456I	VERDADERO	ADCRECV
XC3S400-5FT256I	VERDADERO	ADCRECV
XC3S400-5PQ208I	VERDADERO	ADCRECV
XC3S400-5TQ144I	VERDADERO	ADCRECV
XC3S400A-5FG320I	VERDADERO	ADCRECV
XC3S400A-5FG400I	VERDADERO	ADCRECV
XC3S400A-5FT256I	VERDADERO	ADCRECV
XC3S400AN-5FGG400I	VERDADERO	ADCRECV
XC3S50-5CP132I	VERDADERO	ADCRECV
XC3S50-5PQ208I	VERDADERO	ADCRECV
XC3S50-5TQ144I	VERDADERO	ADCRECV
XC3S50-5VQ100I	VERDADERO	ADCRECV
XC3S500E-5CP132I	VERDADERO	ADCRECV
XC3S500E-5FG320I	VERDADERO	ADCRECV
XC3S500E-5FT256I	VERDADERO	ADCRECV
XC3S500E-5PQ208I	VERDADERO	ADCRECV
XC3S50A-5FT256I	VERDADERO	ADCRECV
XC3S50A-5TQ144I	VERDADERO	ADCRECV
XC3S50A-5VQ100I	VERDADERO	ADCRECV
XC3S50AN-5TQG144I	VERDADERO	ADCRECV
XC3S700A-5FG400I	VERDADERO	ADCRECV
XC3S700A-5FG484I	VERDADERO	ADCRECV
XC3S700A-5FT256I	VERDADERO	ADCRECV
XC3S700AN-5FGG484I	VERDADERO	ADCRECV
XA3S1200E-4FTG256Q	VERDADERO	AES_DEC

Tabla 81. Lista de FPGAs fallidas solo por precio (Balanced)

Tecnologia	Fallo_precio	Fallo_pot
XA3S1200E-4FTG256I	FALSO	VERDADERO
XA3S200A-4FTG256Q	FALSO	VERDADERO
XA3S200A-4FTG256I	FALSO	VERDADERO
XA3S400A-4FGG400I	FALSO	VERDADERO
XA3S400A-4FTG256Q	FALSO	VERDADERO
XA3S400A-4FTG256I	FALSO	VERDADERO
XC3S1400A-4FT256I	FALSO	VERDADERO
XC3S1400A-4FT256C	FALSO	VERDADERO
XC3S1400A-5FT256C	FALSO	VERDADERO
XC3S200A-4FG320I	FALSO	VERDADERO
XC3S200A-4FG320C	FALSO	VERDADERO
XC3S200A-4FT256I	FALSO	VERDADERO
XC3S200A-4FT256C	FALSO	VERDADERO
XC3S200A-4VQ100I	FALSO	VERDADERO
XC3S200A-4VQ100C	FALSO	VERDADERO
XC3S200A-5FG320C	FALSO	VERDADERO
XC3S200A-5FT256C	FALSO	VERDADERO
XC3S200A-5VQ100C	FALSO	VERDADERO
XC3S200AN-4FTG256I	FALSO	VERDADERO
XC3S200AN-4FTG256C	FALSO	VERDADERO
XC3S200AN-5FTG256C	FALSO	VERDADERO
XC3S400A-4FG320I	FALSO	VERDADERO
XC3S400A-4FG320C	FALSO	VERDADERO
XC3S400A-4FG400I	FALSO	VERDADERO
XC3S400A-4FG400C	FALSO	VERDADERO
XC3S400A-4FT256I	FALSO	VERDADERO
XC3S400A-4FT256C	FALSO	VERDADERO
XC3S400A-5FG320C	FALSO	VERDADERO
XC3S400A-5FG400C	FALSO	VERDADERO
XC3S400A-5FT256C	FALSO	VERDADERO
XC3S400AN-4FGG400I	FALSO	VERDADERO
XC3S400AN-4FGG400C	FALSO	VERDADERO
XC3S400AN-5FGG400C	FALSO	VERDADERO
XC3S50A-4FT256I	FALSO	VERDADERO
XC3S50A-4FT256C	FALSO	VERDADERO
XC3S50A-4TQ144I	FALSO	VERDADERO
XC3S50A-4TQ144C	FALSO	VERDADERO
XC3S50A-4VQ100I	FALSO	VERDADERO
XC3S50A-4VQ100C	FALSO	VERDADERO
XC3S50A-5FT256C	FALSO	VERDADERO
XC3S50A-5TQ144C	FALSO	VERDADERO
XC3S50A-5VQ100C	FALSO	VERDADERO

Tecnologia	Fallo_precio	Fallo_pot
XC3S50AN-4TQG144I	FALSO	VERDADERO
XC3S50AN-4TQG144C	FALSO	VERDADERO
XC3S50AN-5TQG144C	FALSO	VERDADERO
XC3S700A-4FG400I	FALSO	VERDADERO
XC3S700A-4FG400C	FALSO	VERDADERO
XC3S700A-4FG484I	FALSO	VERDADERO
XC3S700A-4FG484C	FALSO	VERDADERO
XC3S700A-4FT256I	FALSO	VERDADERO
XC3S700A-4FT256C	FALSO	VERDADERO
XC3S700A-5FG400C	FALSO	VERDADERO
XC3S700A-5FG484C	FALSO	VERDADERO
XC3S700A-5FT256C	FALSO	VERDADERO
XC3S700AN-4FGG484I	FALSO	VERDADERO
XC3S700AN-4FGG484C	FALSO	VERDADERO
XC3S700AN-5FGG484C	FALSO	VERDADERO

Tabla 82. FPGAs-circuito con potencia 0mw (balanced)

Tecnologia	Fallo_precio	Fallo_pot
XA3S1200E-4FTG256Q	VERDADERO	VERDADERO
XA3S400A-4FGG400Q	VERDADERO	VERDADERO
XC3S1400A-5FT256I	VERDADERO	VERDADERO
XC3S200A-5FG320I	VERDADERO	VERDADERO
XC3S200A-5FT256I	VERDADERO	VERDADERO
XC3S200A-5VQ100I	VERDADERO	VERDADERO
XC3S200AN-5FTG256I	VERDADERO	VERDADERO
XC3S400A-5FG320I	VERDADERO	VERDADERO
XC3S400A-5FG400I	VERDADERO	VERDADERO
XC3S400A-5FT256I	VERDADERO	VERDADERO
XC3S400AN-5FGG400I	VERDADERO	VERDADERO
XC3S50A-5FT256I	VERDADERO	VERDADERO
XC3S50A-5TQ144I	VERDADERO	VERDADERO
XC3S50A-5VQ100I	VERDADERO	VERDADERO
XC3S50AN-5TQG144I	VERDADERO	VERDADERO
XC3S700A-5FG400I	VERDADERO	VERDADERO
XC3S700A-5FG484I	VERDADERO	VERDADERO
XC3S700A-5FT256I	VERDADERO	VERDADERO
XC3S700AN-5FGG484I	VERDADERO	VERDADERO

Tabla 83. FPGAs-circuitos con fallos en puntuación por precio superior a 50$ y potencia 0mW (balanced)

A partir de estas tablas, se contabiliza que el máximo total de modelos de FPGAs analizados son los 80 iniciales de Altera más 164 de Xilinx, en este último caso se trata de los 106 iniciales más las modalidades comerciales o industriales de algunas de ellas.

7.4. Normalización de Pk a Pk’

Este apartado hace uso de la tabla *Puntuaciones_max_min*, donde **ResultadosPdC** guarda cada Pk_{max} y PK_{min} por cada uno de los 16 circuitos y cada una de las dos estrategias, devuelve un total de 32 registros, Tabla 84.

Luego mediante una conversión lineal que utiliza los datos de la tabla anterior obtiene los nuevos valores Pk’ que quedan almacenados en *Normalizados*, Tabla 85. Esta nueva tabla cuenta con exactamente el mismo número de registros que puntuaciones, 6.139, y mantiene la información relativa a la estrategia de *tiempo.*

La normalización convierte todas las puntuaciones Pk a valores comprendidos entre 1 y 10, lo que facilita la ordenación de todas las tecnologías frente a cada circuito ensayado.

La Tabla 85 ordena las tecnologías en orden ascendente de P_1’, esta medida relaciona frecuencia y precio y da lugar a que la clasificación ante el circuito SIN sea diferente a la obtenida en la Tabla 80, donde el criterio de ordenación era solo la frecuencia.

Mediante esta clasificación se invierten al menos las 10 primeras posiciones que pasan a ser ocupadas por tecnologías de Xilinx, Altera baja hasta el puesto 29, lo que significa que los productos del primero ofrecen mejor frecuencia por dólar.

Circuito	Estrategia	P1_min	P1_max	P2_min	P2_max	P3_min	P3_max	P4_min	P4_max	P5_min	P5_max
ADCRECV	BALANCED	3,24427	47,9782	1,999066	17,57555	2,004008	18,11594	2,004008	18,11594	0,2057808	27,63958
AES_DEC	BALANCED	1,440381	11,54232	0,0896884	5,444267	1,726886	7,911393	1,988103	10,54852	0,1250125	3,192848
AES_ENC	BALANCED	2,126031	12,15688	0,7687577	5,63713	1,726886	7,911393	1,988103	10,54852	0,1257748	3,192848
AES128_FAST	BALANCED	1,224204	2,094314	0,7194078	1,136555	2,004008	3,012048	2,004008	3,012048	0,1678895	0,2868617
ANALYTIC_FILTER_H_A1	BALANCED	3,851612	24,1385	1,973963	9,032173	2,003678	10,75269	1,859724	7,810934	0,1281184	3,468609
DDS_SYNTHESIZER	BALANCED	3,301387	28,42686	1,990872	16,70063	2,004008	18,11594	2,004008	18,11594	0,1273275	12,93996
FPU	BALANCED	0,3524066	4,886356	0,05764597	2,509072	2,004008	7,194245	1,766285	7,194245	0,1079951	1,357405
I2CSLAVETOP	BALANCED	2,621898	26,06488	1,975264	13,94722	2,004008	18,11594	2,004008	18,11594	0,1289191	15,09662
RS_DEC	BALANCED	1,95647	8,20727	0,0649104	2,794432	0,6641871	7,281464	1,87578	7,281464	0,1199248	1,860119
RS_ENCODE	BALANCED	2,573253	5,951165	1,694078	3,71769	2,004008	4,56621	2,004008	4,56621	0,1854084	1,268392
SC_CORPROC	BALANCED	3,04542	26,95424	1,858339	8,311717	2,004008	18,11594	2,004008	18,11594	0,1273275	13,93534
SIN	BALANCED	2,548134	24,16426	1,998805	17,70422	2,004008	18,11594	2,004008	18,11594	0,1178689	15,09662
SPI_BOOT	BALANCED	2,699877	26,4312	1,990481	16,41757	2,004008	18,11594	2,004008	18,11594	0,1273275	13,93534
T400_CORE	BALANCED	1,483428	10,23861	1,847927	6,187238	2,004008	13,44086	2,004008	13,44086	0,1171992	7,074137
UART_16750	BALANCED	2,085438	20,16467	1,866505	8,620512	1,976696	12,0773	1,999286	18,11594	0,1185463	12,93996
UCRC_PAR	BALANCED	4,833747	48,4254	2,000886	17,83288	2,004008	18,11594	2,004008	18,11594	0,1281184	15,09662
ADCRECV	TIEMPO	4,347779	58,90947	1,999326	17,34396	2,004008	18,11594	2,004008	18,11594	0,1250125	12,07729
AES_DEC	TIEMPO	1,838417	13,93096	0,1304121	5,252849	1,726886	7,911393	1,988103	10,54852	0,1250125	2,693965
AES_ENC	TIEMPO	2,195433	14,29074	0,334181	5,377916	1,726886	7,911393	1,988103	10,54852	0,1257748	2,693965
AES128_FAST	TIEMPO	1,54619	2,628766	0,6822073	1,086902	2,004008	3,012048	2,004008	3,012048	0,1686085	0,28962
ANALYTIC_FILTER_H_A1	TIEMPO	4,751685	24,74437	1,959657	8,856365	2,004008	10,75269	1,859724	7,8125	0,1273275	3,468609
DDS_SYNTHESIZER	TIEMPO	3,311673	34,45406	1,990872	16,70063	2,004008	18,11594	2,004008	18,11594	0,1273275	12,93996
FPU	TIEMPO	0,3865182	4,973625	0,09704464	2,431236	2,004008	7,194245	1,766285	7,194245	0,09964765	1,357405
I2CSLAVETOP	TIEMPO	3,01176	27,61998	1,975524	13,2267	2,004008	18,11594	2,004008	18,11594	0,1289191	15,09662
RS_DEC	TIEMPO	1,750488	10,13797	0,09669424	2,546633	0,6641871	7,281464	1,87578	7,281464	0,1185463	1,753648

Circuito	Estrategia	P1_min	P1_max	P2_min	P2_max	P3_min	P3_max	P4_min	P4_max	P5_min	P5_max
RS_ENCODE	TIEMPO	2,710346	6,784049	1,618025	3,550789	2,004008	4,56621	2,004008	4,56621	0,1854084	1,11371
SC_CORPROC	TIEMPO	3,088344	29,38515	1,836316	7,513999	2,004008	18,11594	2,004008	18,11594	0,1250125	13,93534
SIN	TIEMPO	2,74126	24,70893	1,998805	17,70422	2,004008	18,11594	2,004008	18,11594	0,1178689	15,09662
SPI_BOOT	TIEMPO	3,079126	31,79351	1,990221	16,26318	2,004008	18,11594	2,004008	18,11594	0,1257748	12,07729
T400_CORE	TIEMPO	1,587104	10,79803	1,818899	6,082634	2,004008	13,44086	2,004008	13,44086	0,1171992	7,074137
UART_16750	TIEMPO	2,222988	22,50707	1,834574	8,002923	1,976696	12,0773	1,999286	18,11594	0,1171992	12,93996
UCRC_PAR	TIEMPO	5,017856	48,92234	2,000886	17,80715	2,004008	18,11594	2,004008	18,11594	0,1281184	15,09662

Tabla 84. Contenido de la tabla Puntuaciones_max_min

Circuito	Estrategia	Tecnologia	P1_N	P2_N	P3_N	P4_N	P5_N
SIN	BALANCED	XC3S50A-4VQ100C	10,0000	10,0000	10,0000	10,0000	10,0000
SIN	BALANCED	XC3S50A-5VQ100C	9,6339	8,3398	8,3441	8,3441	8,5157
SIN	BALANCED	XC3S50A-4VQ100I	8,6712	8,6600	8,6635	8,6635	8,8020
SIN	BALANCED	XC3S50A-5TQ144C	7,2767	6,1189	6,1288	6,1288	6,5300
SIN	BALANCED	XC3S50A-4TQ144C	7,1833	7,3818	7,3885	7,3885	7,6591
SIN	BALANCED	XC3S50A-4TQ144I	6,2208	6,3817	6,3910	6,3910	6,7650
SIN	BALANCED	XC3S50A-5FT256C	6,1511	5,1579	5,1703	5,1703	5,6707
SIN	BALANCED	XC3S50A-4FT256C	6,1366	6,2476	6,2572	6,2572	6,6450
SIN	BALANCED	XC3S50AN-5TQG144C	6,0470	5,0690	5,0816	5,0816	4,9253
SIN	BALANCED	XC3S50AN-4TQG144C	5,9678	6,1189	6,1288	6,1288	5,7298
SIN	BALANCED	XC3S100E-4VQ100C	5,6862	5,7987	5,7729	5,7729	2,2227
SIN	BALANCED	XC3S100E-5VQ100C	5,6709	5,0148	4,9959	4,9959	2,0521
SIN	BALANCED	XC3S50-4TQ144C	5,3835	5,8880	5,8869	5,8869	2,4317
SIN	BALANCED	XC3S50-5TQ144C	5,3653	5,0986	5,1011	5,1011	2,2351
SIN	BALANCED	XC3S50A-4FT256I	5,3251	5,4104	5,4222	5,4222	5,8965
SIN	BALANCED	XC3S50AN-4TQG144I	5,1617	5,2812	5,2933	5,2933	5,0879
SIN	BALANCED	XC3S100E-5TQ144C	5,1152	4,2638	4,2514	4,2514	1,8887
SIN	BALANCED	XC3S50-5VQ100C	5,0715	5,4101	5,4112	5,4112	2,3127
SIN	BALANCED	XC3S100E-4VQ100I	4,9283	5,0148	4,9959	4,9959	2,0521
SIN	BALANCED	XC3S100E-4CP132I	4,8752	4,9312	4,9129	4,9129	2,0339
SIN	BALANCED	XC3S100E-4CP132C	4,8752	4,9312	4,9129	4,9129	2,0339
SIN	BALANCED	XC3S100E-5CP132C	4,8633	4,2638	4,2514	4,2514	1,8887
SIN	BALANCED	XC3S50-4CP132C	4,8196	5,8880	5,8869	5,8869	2,4317
SIN	BALANCED	XC3S200A-4VQ100C	4,7369	4,7506	4,6960	4,6960	3,1812
SIN	BALANCED	XC3S200A-5VQ100C	4,7346	4,0803	4,0368	4,0368	2,8729

Circuito	Estrategia	Tecnologia	P1_N	P2_N	P3_N	P4_N	P5_N
SIN	BALANCED	XC3S50-5CP132C	4,7147	5,0986	5,1011	5,1011	2,2351
SIN	BALANCED	XC3S50-4TQ144I	4,6711	5,0986	5,1011	5,1011	2,2351
SIN	BALANCED	EP3C5E144C7	4,5853	3,5708	3,5315	3,5315	1,5837
SIN	BALANCED	EP3C5E144C8	4,5727	4,2967	4,2446	4,2446	1,7115
SIN	BALANCED	XC3S200-4VQ100C	4,5546	4,7116	4,6549	4,6549	1,8301
SIN	BALANCED	XC3S200-5VQ100C	4,5511	4,0797	4,0337	4,0337	1,7129
SIN	BALANCED	XC3S100E-4TQ144I	4,5328	4,2638	4,2514	4,2514	1,8887
SIN	BALANCED	XC3S100E-4TQ144C	4,5328	4,2638	4,2514	4,2514	1,8887
SIN	BALANCED	XA3S100E-4VQG100I	4,4567	4,5271	4,5124	4,5124	1,9460
SIN	BALANCED	XC3S50-4VQ100I	4,3956	5,4101	5,4112	5,4112	2,3127
SIN	BALANCED	XC3S50-4VQ100C	4,3956	5,4101	5,4112	5,4112	2,3127
SIN	BALANCED	XC3S250E-4VQ100C	4,3091	4,2071	4,1512	4,1512	1,6047
SIN	BALANCED	XC3S250E-5VQ100C	4,2990	3,6348	3,5897	3,5897	1,5159
SIN	BALANCED	XC3S200A-4VQ100I	4,2736	4,2777	4,2310	4,2310	2,9638
SIN	BALANCED	XC3S50-4PQ208C	4,2462	4,4919	4,4970	4,4970	2,0840
SIN	BALANCED	XC3S50-4CP132I	4,1811	5,0986	5,1011	5,1011	2,2351
SIN	BALANCED	EP2C5T144C7	4,1064	3,5675	3,5315	3,5315	1,9906
SIN	BALANCED	XA3S100E-4TQG144I	4,1027	3,8510	3,8422	3,8422	1,7988
SIN	BALANCED	EP3C5F256C8	4,0874	3,6706	3,6295	3,6295	1,6013
SIN	BALANCED	XC3S200A-4FT256C	3,9649	4,0803	4,0368	4,0368	2,8729
SIN	BALANCED	XC3S200-4VQ100I	3,9541	4,0797	4,0337	4,0337	1,7129
SIN	BALANCED	EP3C5M164C7	3,9220	3,0311	3,0012	3,0012	1,4886
SIN	BALANCED	EP2C5T144C6	3,8948	2,8133	2,7899	2,7899	1,7750
SIN	BALANCED	XC3S200A-5FT256C	3,8793	3,3734	3,3415	3,3415	2,5478
SIN	BALANCED	XA3S100E-4VQG100Q	3,8700	3,9202	3,9108	3,9108	1,8139

Circuito	Estrategia	Tecnologia	P1_N	P2_N	P3_N	P4_N	P5_N
SIN	BALANCED	XC3S50-5PQ208C	3,8449	3,8914	3,8992	3,8992	1,9345
SIN	BALANCED	XA3S100E-4CPG132I	3,8249	3,8510	3,8422	3,8422	1,7988
SIN	BALANCED	EP3C5F256C7	3,7933	3,0311	3,0012	3,0012	1,4886
SIN	BALANCED	EP2C5Q208C8	3,7626	3,9415	3,8992	3,8992	2,0975
SIN	BALANCED	XC3S250E-4VQ100I	3,7346	3,6348	3,5897	3,5897	1,5159
SIN	BALANCED	XC3S50-4PQ208I	3,6884	3,8914	3,8992	3,8992	1,9345
SIN	BALANCED	EP2C5Q208C7	3,6882	3,2562	3,2254	3,2254	1,9016
SIN	BALANCED	EP3C5U256C8	3,6852	3,3006	3,2660	3,2660	1,5361
SIN	BALANCED	XC3S250E-4TQ144C	3,6648	3,5755	3,5315	3,5315	1,5067
SIN	BALANCED	XC3S250E-5TQ144C	3,6641	3,0930	3,0580	3,0580	1,4318
SIN	BALANCED	XC3S250E-4CP132C	3,6475	3,5755	3,5315	3,5315	1,5067
SIN	BALANCED	XC3S250E-5CP132C	3,6466	3,0930	3,0580	3,0580	1,4318
SIN	BALANCED	EP2C5F256C8	3,5999	3,6672	3,6295	3,6295	2,0191
SIN	BALANCED	XA3S100E-4TQG144Q	3,5496	3,3201	3,3160	3,3160	1,6833
SIN	BALANCED	EP2C5T144C8	3,5437	4,2927	4,2446	4,2446	2,1979
SIN	BALANCED	XC3S400A-5FT256C	3,5224	2,8540	2,8174	2,8174	2,0992
SIN	BALANCED	XC3S200-5TQ144C	3,5172	3,6180	3,5799	3,5799	1,6273
SIN	BALANCED	XC3S400A-4FT256C	3,4834	3,4562	3,4070	3,4070	2,3341
SIN	BALANCED	EP2C5F256C7	3,4532	3,0282	3,0012	3,0012	1,8364
SIN	BALANCED	XC3S200A-4FT256I	3,4480	3,5377	3,5031	3,5031	2,6233
SIN	BALANCED	EP3C5U256C7	3,4411	2,7408	2,7161	2,7161	1,4375
SIN	BALANCED	XC3S200A-4FG320C	3,3863	3,4537	3,4205	3,4205	2,5847
SIN	BALANCED	XA3S250E-4VQG100I	3,3659	3,2677	3,2295	3,2295	1,4589
SIN	BALANCED	XC3S200AN-4FTG256C	3,3549	3,4400	3,4070	3,4070	2,3341
SIN	BALANCED	XA3S100E-4CPG132Q	3,3087	3,3201	3,3160	3,3160	1,6833

Circuito	Estrategia	Tecnologia	P1_N	P2_N	P3_N	P4_N	P5_N
SIN	BALANCED	XC3S200AN-5FTG256C	3,2932	2,8500	2,8267	2,8267	2,1029
SIN	BALANCED	EP3C5E144I7	3,2402	2,8160	2,7899	2,7899	1,4508
SIN	BALANCED	EP2C5F256C6	3,2353	2,3906	2,3743	2,3743	1,6541
SIN	BALANCED	XC3S250E-4PQ208C	3,1884	2,9655	2,9330	2,9330	1,4120
SIN	BALANCED	XC3S200-4TQ144I	3,1841	3,6180	3,5799	3,5799	1,6273
SIN	BALANCED	XC3S200-4TQ144C	3,1841	3,6180	3,5799	3,5799	1,6273
SIN	BALANCED	XC3S250E-5PQ208C	3,1822	2,5578	2,5330	2,5330	1,3487
SIN	BALANCED	XC3S250E-4TQ144I	3,1816	3,0930	3,0580	3,0580	1,4318
SIN	BALANCED	EP3C10E144C8	3,1713	2,8277	2,7899	2,7899	1,4508
SIN	BALANCED	XC3S250E-4CP132I	3,1666	3,0930	3,0580	3,0580	1,4318
SIN	BALANCED	XC3S200A-5FG320C	3,1332	2,8500	2,8267	2,8267	2,3070
SIN	BALANCED	EP2C5Q208I8	3,1215	3,2562	3,2254	3,2254	1,9016
SIN	BALANCED	XC3S400A-4FT256I	3,0171	2,9823	2,9430	2,9430	2,1492
SIN	BALANCED	EP3C5F256C6	3,0112	1,9762	1,9649	1,9649	1,3028
SIN	BALANCED	EP2C5F256I8	2,9864	3,0282	3,0012	3,0012	1,8364
SIN	BALANCED	EP2C5T144I8	2,9547	3,5675	3,5315	3,5315	1,9906
SIN	BALANCED	XC3S200A-4FG320I	2,9411	2,9889	2,9633	2,9633	2,3709
SIN	BALANCED	XA3S250E-4VQG100Q	2,9162	2,8197	2,7899	2,7899	1,3894
SIN	BALANCED	XC3S200AN-4FTG256I	2,9153	2,9785	2,9531	2,9531	2,1533
SIN	BALANCED	XC3S400A-4FG320C	2,8936	2,9119	2,8741	2,8741	2,1218
SIN	BALANCED	EP2C8T144C8	2,8882	2,8248	2,7899	2,7899	1,5678
SIN	BALANCED	XA3S250E-4TQG144I	2,8623	2,7741	2,7452	2,7452	1,3823
SIN	BALANCED	XA3S250E-4CPG132I	2,8488	2,7741	2,7452	2,7452	1,3823
SIN	BALANCED	EP3C10E144C7	2,8350	2,3257	2,2987	2,2987	1,3627
SIN	BALANCED	EP2C8T144C7	2,7686	2,3233	2,2987	2,2987	1,4711

Circuito	Estrategia	Tecnologia	P1_N	P2_N	P3_N	P4_N	P5_N
SIN	BALANCED	XC3S250E-4PQ208I	2,7625	2,5578	2,5330	2,5330	1,3487
SIN	BALANCED	XC3S500E-4CP132C	2,7355	2,6055	2,5713	2,5713	1,2276
SIN	BALANCED	XC3S400A-5FG320C	2,7343	2,2947	2,2698	2,2698	1,8810
SIN	BALANCED	XC3S500E-5CP132C	2,7336	2,2460	2,2197	2,2197	1,1886
SIN	BALANCED	EP3C5U256C6	2,7206	1,7755	1,7677	1,7677	1,2675
SIN	BALANCED	EP2C8Q208C7	2,6553	2,2014	2,1793	2,1793	1,4338
SIN	BALANCED	XC3S250E-4FT256C	2,6378	2,6701	2,6431	2,6431	1,3662
SIN	BALANCED	XC3S250E-5FT256C	2,6313	2,3001	2,2800	2,2800	1,3087
SIN	BALANCED	EP2C8T144C6	2,6034	1,8347	1,8201	1,8201	1,3638
SIN	BALANCED	EP2C8Q208C8	2,5955	2,6638	2,6323	2,6323	1,5332
SIN	BALANCED	EP3C10F256C8	2,5701	2,2836	2,2576	2,2576	1,3553
SIN	BALANCED	EP3C5F256I7	2,5374	1,9762	1,9649	1,9649	1,3028
SIN	BALANCED	XC3S200-5PQ208C	2,5302	2,6609	2,6391	2,6391	1,4497
SIN	BALANCED	XC3S500E-4VQ100C	2,5262	2,6055	2,5713	2,5713	1,2276
SIN	BALANCED	EP3C5E144A7	2,5224	2,0083	1,9965	1,9965	1,3085
SIN	BALANCED	XC3S400A-4FG320I	2,5129	2,5179	2,4884	2,4884	1,9681
SIN	BALANCED	XA3S250E-4PQG208I	2,4867	2,2938	2,2739	2,2739	1,3078
SIN	BALANCED	XC3S400A-4FG400C	2,4838	2,4596	2,4312	2,4312	1,9453
SIN	BALANCED	XA3S250E-4TQG144Q	2,4793	2,3916	2,3698	2,3698	1,3229
SIN	BALANCED	XA3S250E-4CPG132Q	2,4676	2,3916	2,3698	2,3698	1,3229
SIN	BALANCED	EP2C8T144I8	2,3903	2,3233	2,2987	2,2987	1,4600
SIN	BALANCED	XC3S500E-4CP132I	2,3701	2,2460	2,2197	2,2197	1,1886
SIN	BALANCED	XC3S700A-4FT256C	2,3432	2,3946	2,3632	2,3632	1,7384
SIN	BALANCED	EP3C10F256C7	2,3425	1,8860	1,8684	1,8684	1,2856
SIN	BALANCED	EP3C10U256C8	2,3355	2,0671	2,0457	2,0457	1,3173

Circuito	Estrategia	Tecnologia	P1_N	P2_N	P3_N	P4_N	P5_N
SIN	BALANCED	XA3S500E-4CPG132I	2,3222	2,1989	2,1736	2,1736	1,1835
SIN	BALANCED	XC3S400-4TQ144C	2,3152	2,8415	2,8051	2,8051	1,3431
SIN	BALANCED	XC3S700A-5FT256C	2,2937	1,9712	1,9494	1,9494	1,6035
SIN	BALANCED	EP3C5U256I7	2,2916	1,7755	1,7677	1,7677	1,2675
SIN	BALANCED	XC3S400-5TQ144C	2,2910	2,4491	2,4210	2,4210	1,2887
SIN	BALANCED	XC3S250E-4FT256I	2,2831	2,3001	2,2800	2,2800	1,3087
SIN	BALANCED	EP3C16E144C8	2,2773	1,9953	1,9727	1,9727	1,2553
SIN	BALANCED	EP2C8F256C7	2,2735	1,8840	1,8684	1,8684	1,3746
SIN	BALANCED	EP3C10F256C6	2,2643	1,4763	1,4675	1,4675	1,2137
SIN	BALANCED	XC3S400A-5FG400C	2,2492	2,0254	2,0061	2,0061	1,7760
SIN	BALANCED	EP3C10E144I7	2,2382	1,8366	1,8201	1,8201	1,2769
SIN	BALANCED	XC3S200-4PQ208I	2,2110	2,6609	2,6391	2,6391	1,4497
SIN	BALANCED	XC3S200-4PQ208C	2,2110	2,6609	2,6391	2,6391	1,4497
SIN	BALANCED	EP2C8F256C8	2,2064	2,2813	2,2576	2,2576	1,4618
SIN	BALANCED	XC3S500E-4PQ208C	2,1962	2,1481	2,1239	2,1239	1,1779
SIN	BALANCED	XC3S500E-5PQ208C	2,1956	1,8500	1,8323	1,8323	1,1456
SIN	BALANCED	XC3S500E-4VQ100I	2,1882	2,2460	2,2197	2,2197	1,1886
SIN	BALANCED	EP3C16Q240C8	2,1764	1,8187	1,8001	1,8001	1,2284
SIN	BALANCED	EP2C8Q208I8	2,1582	2,2014	2,1793	2,1793	1,4338
SIN	BALANCED	XA3S250E-4PQG208Q	2,1562	1,9773	1,9633	1,9633	1,2586
SIN	BALANCED	XC3S400AN-4FGG400C	2,1558	2,1238	2,1025	2,1025	1,7002
SIN	BALANCED	XC3S400A-4FG400I	2,1506	2,1184	2,0972	2,0972	1,8123
SIN	BALANCED	EP3C16E144C7	2,1328	1,6407	1,6262	1,6262	1,2013
SIN	BALANCED	EP3C10U256C7	2,1247	1,7019	1,6883	1,6883	1,2533
SIN	BALANCED	EP3C16M164C8	2,1169	1,8601	1,8405	1,8405	1,2347

Circuito	Estrategia	Tecnologia	P1_N	P2_N	P3_N	P4_N	P5_N
SIN	BALANCED	EP2C8F256C6	2,1114	1,4747	1,4675	1,4675	1,2848
SIN	BALANCED	XC3S400-5PQ208C	2,0555	2,0962	2,0754	2,0754	1,2398
SIN	BALANCED	XC3S500E-4FT256C	2,0542	1,9373	1,9177	1,9177	1,1551
SIN	BALANCED	XA3S250E-4FTG256I	2,0541	2,0612	2,0457	2,0457	1,2717
SIN	BALANCED	EP3C10U256C6	2,0540	1,3296	1,3240	1,3240	1,1879
SIN	BALANCED	XC3S200-4FT256C	2,0532	2,0788	2,0668	2,0668	1,3418
SIN	BALANCED	XC3S200-4FT256I	2,0532	2,0787	2,0668	2,0668	1,3418
SIN	BALANCED	XC3S500E-5FT256C	2,0499	1,6641	1,6505	1,6505	1,1255
SIN	BALANCED	EP3C16F256C7	2,0492	1,4646	1,4541	1,4541	1,1745
SIN	BALANCED	XC3S200-5FT256C	2,0485	2,0787	2,0668	2,0668	1,3418
SIN	BALANCED	EP3C16F256C8	2,0448	1,7855	1,7677	1,7677	1,2234
SIN	BALANCED	XC3S700A-4FT256I	2,0308	2,0645	2,0406	2,0406	1,6333
SIN	BALANCED	EP3C16M164C7	2,0194	1,5209	1,5091	1,5091	1,1831
SIN	BALANCED	XC3S400-5FT256C	2,0145	1,9675	1,9494	1,9494	1,2259
SIN	BALANCED	XA3S500E-4CPG132Q	2,0109	1,8927	1,8741	1,8741	1,1502
SIN	BALANCED	XC3S400-4TQ144I	2,0067	2,4536	2,4254	2,4254	1,2894
SIN	BALANCED	XC3S400AN-5FGG400C	1,9507	1,7449	1,7314	1,7314	1,5714
SIN	BALANCED	EP3C16U256C7	1,9104	1,3587	1,3506	1,3506	1,1583
SIN	BALANCED	EP3C16U256C8	1,9053	1,6576	1,6427	1,6427	1,2039
SIN	BALANCED	XC3S500E-4PQ208I	1,9029	1,8500	1,8323	1,8323	1,1456
SIN	BALANCED	XC3S400AN-4FGG400I	1,8661	1,8272	1,8121	1,8121	1,5994
SIN	BALANCED	XA3S500E-4PQG208I	1,8625	1,8090	1,7923	1,7923	1,1412
SIN	BALANCED	EP2C15AF256C6	1,8490	1,3236	1,3165	1,3165	1,1351
SIN	BALANCED	EP3C10F256I7	1,8410	1,4763	1,4675	1,4675	1,2137
SIN	BALANCED	EP2C8AF256I8	1,8353	1,8840	1,8684	1,8684	1,3746

Circuito	Estrategia	Tecnologia	P1_N	P2_N	P3_N	P4_N	P5_N
SIN	BALANCED	EP2C8F256I8	1,8353	1,8840	1,8684	1,8684	1,3746
SIN	BALANCED	EP3C16F484C8	1,8326	1,5762	1,5631	1,5631	1,1915
SIN	BALANCED	EP3C16F256C6	1,8191	1,1419	1,1387	1,1387	1,1253
SIN	BALANCED	XC3S400-4PQ208I	1,8053	2,0962	2,0754	2,0754	1,2398
SIN	BALANCED	XC3S400-4PQ208C	1,8053	2,0962	2,0754	2,0754	1,2398
SIN	BALANCED	XC3S700A-4FG400C	1,8042	1,7370	1,7205	1,7205	1,5289
SIN	BALANCED	XA3S250E-4FTG256Q	1,7801	1,7753	1,7652	1,7652	1,2273
SIN	BALANCED	XC3S500E-4FT256I	1,7767	1,6641	1,6505	1,6505	1,1255
SIN	BALANCED	XA3S200A-4FTG256I	1,7683	1,7746	1,7690	1,7690	1,8123
SIN	BALANCED	XA3S500E-4FTG256I	1,7424	1,6303	1,6175	1,6175	1,1218
SIN	BALANCED	XC3S500E-4FG320C	1,7403	1,6641	1,6505	1,6505	1,1255
SIN	BALANCED	XC3S500E-5FG320C	1,7380	1,4278	1,4194	1,4194	1,0998
SIN	BALANCED	EP3C16F484C7	1,7195	1,2871	1,2806	1,2806	1,1474
SIN	BALANCED	XC3S400-4FT256I	1,7087	1,9675	1,9494	1,9494	1,2259
SIN	BALANCED	XC3S400-4FT256C	1,7087	1,9675	1,9494	1,9494	1,2259
SIN	BALANCED	EP3C16U256C6	1,6964	1,0579	1,0566	1,0566	1,1125
SIN	BALANCED	EP3C10U256I7	1,6690	1,3296	1,3240	1,3240	1,1879
SIN	BALANCED	EP3C16U484C8	1,6614	1,4205	1,4110	1,4110	1,1678
SIN	BALANCED	XA3S500E-4PQG208Q	1,6118	1,5542	1,5431	1,5431	1,1135
SIN	BALANCED	XC3S700A-5FG400C	1,6023	1,4212	1,4118	1,4118	1,4283
SIN	BALANCED	XC3S1400A-4FT256C	1,5818	1,5307	1,5168	1,5168	1,2379
SIN	BALANCED	EP3C10E144A7	1,5751	1,2997	1,2947	1,2947	1,1827
SIN	BALANCED	XA3S200A-4FTG256Q	1,5727	1,5694	1,5671	1,5671	1,7179
SIN	BALANCED	XC3S700A-4FG400I	1,5606	1,4912	1,4802	1,4802	1,4506
SIN	BALANCED	EP3C16U484C7	1,5573	1,1565	1,1530	1,1530	1,1275

Circuito	Estrategia	Tecnologia	P1_N	P2_N	P3_N	P4_N	P5_N
SIN	BALANCED	XA3S400A-4FTG256I	1,5570	1,4987	1,4904	1,4904	1,5705
SIN	BALANCED	XC3S700A-5FG484C	1,5560	1,2657	1,2598	1,2598	1,3787
SIN	BALANCED	XC3S700A-4FG484C	1,5526	1,5464	1,5342	1,5342	1,4682
SIN	BALANCED	XC3S1400A-5FT256C	1,5505	1,2510	1,2437	1,2437	1,1864
SIN	BALANCED	EP3C25E144C8	1,5337	1,3030	1,2947	1,2947	1,0876
SIN	BALANCED	EP3C16F484C6	1,5128	1,0000	1,0000	1,0000	1,1037
SIN	BALANCED	XA3S500E-4FTG256Q	1,5082	1,3997	1,3919	1,3919	1,0968
SIN	BALANCED	XC3S400-4FG320C	1,5081	1,4526	1,4453	1,4453	1,1536
SIN	BALANCED	XC3S400-5FG320C	1,5065	1,2443	1,2413	1,2413	1,1243
SIN	BALANCED	XC3S500E-4FG320I	1,5051	1,4278	1,4194	1,4194	1,0998
SIN	BALANCED	EP2C15AF256C7	1,4748	1,2283	1,2233	1,2233	1,1218
SIN	BALANCED	EP2C15AF256C8	1,4689	1,5015	1,4904	1,4904	1,1601
SIN	BALANCED	XC3S1200E-4FT256C	1,4652	1,3582	1,3490	1,3490	1,0194
SIN	BALANCED	XC3S1200E-5FT256C	1,4648	1,1623	1,1577	1,1577	1,0077
SIN	BALANCED	XC3S700AN-5FGG484C	1,4080	1,1365	1,1336	1,1336	1,2839
SIN	BALANCED	XC3S700AN-4FGG484C	1,4066	1,3933	1,3846	1,3846	1,3549
SIN	BALANCED	EP3C25Q240C8	1,3944	1,1944	1,1888	1,1888	1,0758
SIN	BALANCED	XA3S400A-4FTG256Q	1,3831	1,3219	1,3173	1,3173	1,5015
SIN	BALANCED	XC3S1400A-4FT256I	1,3661	1,3107	1,3019	1,3019	1,1974
SIN	BALANCED	XC3S1000-5FT256I	1,3621	1,1572	1,1530	1,1530	1,0492
SIN	BALANCED	EP2C15AF484C8	1,3547	1,3236	1,3165	1,3165	1,1351
SIN	BALANCED	XC3S700A-4FG484I	1,3431	1,3267	1,3195	1,3195	1,3982
SIN	BALANCED	EP3C10F256A7	1,3430	1,0364	1,0371	1,0371	1,1365
SIN	BALANCED	EP3C25F256C8	1,3368	1,1316	1,1274	1,1274	1,0689
SIN	BALANCED	EP2C15AF484C7	1,3276	1,0783	1,0767	1,0767	1,1007

Circuito	Estrategia	Tecnologia	P1_N	P2_N	P3_N	P4_N	P5_N
SIN	BALANCED	XC3S400-4FG320I	1,3036	1,2443	1,2413	1,2413	1,1243
SIN	BALANCED	XC3S1200E-4FG320C	1,2981	1,1623	1,1577	1,1577	1,0081
SIN	BALANCED	XC3S1000-5FT256C	1,2796	1,0817	1,0793	1,0793	1,0423
SIN	BALANCED	EP2C20F256C8	1,2748	1,1938	1,1888	1,1888	1,1168
SIN	BALANCED	EP3C25U256C8	1,2742	1,0745	1,0716	1,0716	1,0626
SIN	BALANCED	XC3S1200E-4FT256I	1,2663	1,1623	1,1577	1,1577	1,0077
SIN	BALANCED	EP2C20Q240C8	1,2580	1,1938	1,1888	1,1888	1,1168
SIN	BALANCED	EP2C15AF256I8	1,2152	1,2283	1,2233	1,2233	1,1218
SIN	BALANCED	XC3S700AN-4FGG484I	1,2149	1,1923	1,1881	1,1881	1,2993
SIN	BALANCED	EP3C25F324C8	1,2024	1,0151	1,0136	1,0136	1,0561
SIN	BALANCED	XC3S1000-4FG320C	1,1773	1,1845	1,1796	1,1796	1,0518
SIN	BALANCED	EP2C20F484C8	1,1607	1,0738	1,0716	1,0716	1,1023
SIN	BALANCED	XA3S1200E-4FTG256I	1,1365	1,0344	1,0328	1,0328	1,0000
SIN	BALANCED	EP2C15AF484I8	1,1182	1,0783	1,0767	1,0767	1,1007
SIN	BALANCED	XC3S400-5FG456C	1,1147	1,0671	1,0678	1,0678	1,0995
SIN	BALANCED	XC3S1000-4FT256I	1,1053	1,0817	1,0793	1,0793	1,0423
SIN	BALANCED	XC3S1000-4FT256C	1,1053	1,0817	1,0793	1,0793	1,0423
SIN	BALANCED	XA3S400A-4FGG400I	1,0987	1,0416	1,0429	1,0429	1,3922
SIN	BALANCED	XC3S1000-4FG320I	1,0676	1,0667	1,0645	1,0645	1,0409
SIN	BALANCED	XC3S400-4FG456I	1,0000	1,0671	1,0678	1,0678	1,0995
SIN	BALANCED	XC3S400-4FG456C	1,0000	1,0671	1,0678	1,0678	1,0995

Tabla 85. Muestra de 246 registros de la tabla Normalizados

7.5. Circuitos implementados

Otra tarea de **ResultadosPdC** consiste en devolver una tabla con las 246 FPGA analizadas y junto a ellas, la lista de circuitos del PdC que se adecuaron correctamente a ellas, se presentan solo algunos de los registos en la Tabla 87.

A partir de la Tabla 87, se realizan los cálculos oportunos para resumir en la Tabla 86, la tasa de éxito de los circuitos del PdC en el conjunto de tecnologías utilizadas:

Circuitos	Total Altera	%Altera	Total Xilinx	%Xilinx	Total FPGAs
ADCRECV	80	100,00%	165	99,40%	245
AES128_FAST	9	11,25%	0	0,00%	9
AES_DEC	80	100,00%	142	85,54%	222
AES_ENC	80	100,00%	142	85,54%	222
SIN	80	100,00%	166	100,00%	246
SPI_BOOT	80	100,00%	166	100,00%	246
T400_CORE	59	73,75%	134	80,72%	193
ANALYTIC_FILTER_H_A1	80	100,00%	110	66,27%	190
DDS_SYNTHESIZER	80	100,00%	166	100,00%	246
I2CSLAVETOP	80	100,00%	166	100,00%	246
FPU	58	72,50%	26	15,66%	84
RS_DEC	80	100,00%	76	45,78%	156
RS_ENCODE	9	11,25%	16	9,64%	25
UART_16750	80	100,00%	166	100,00%	246
UCRC_PAR	80	100,00%	166	100,00%	246
SC_CORPROC	80	100,00%	166	100,00%	246
		85,55%		74,28%	

Tabla 86. Circuitos implementados con éxito por fabricante

Se destacan en la Tabla 86 los siete circuitos que se alojaron en todas las FPGAs de la muestra. Se observa en la Tabla 86, que el conjunto de circuitos ha funcionado mejor con las tecnologías seleccionadas del fabricante Altera que con las de Xilinx.

Tecnología	Circuitos_implementados
XC3S200A-4FG320C	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,DDS_SYNTHESIZER,I2CSLAVETOP,UART_16750,UCRC_PAR,SC_CORPROC
XC3S200A-4FT256I	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,DDS_SYNTHESIZER,I2CSLAVETOP,UART_16750,UCRC_PAR,SC_CORPROC
XC3S200A-4FT256C	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,DDS_SYNTHESIZER,I2CSLAVETOP,UART_16750,UCRC_PAR,SC_CORPROC
XC3S200A-5FG320C	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,DDS_SYNTHESIZER,I2CSLAVETOP,UART_16750,UCRC_PAR,SC_CORPROC
XC3S200A-5FT256C	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,DDS_SYNTHESIZER,I2CSLAVETOP,UART_16750,UCRC_PAR,SC_CORPROC
XC3S200AN-4FTG256I	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,DDS_SYNTHESIZER,I2CSLAVETOP,UART_16750,UCRC_PAR,SC_CORPROC
XC3S200AN-4FTG256C	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,DDS_SYNTHESIZER,I2CSLAVETOP,UART_16750,UCRC_PAR,SC_CORPROC
XC3S200AN-5FTG256C	ADCRECV,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,DDS_SYNTHESIZER,I2CSLAVETOP,UART_16750,UCRC_PAR,SC_CORPROC
EP2C15AF484C7	ADCRECV,AES128_FAST,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,RS_ENCODE,UART_16750,UCRC_PAR,SC_CORPROC
EP2C15AF484C8	ADCRECV,AES128_FAST,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,RS_ENCODE,UART_16750,UCRC_PAR,SC_CORPROC
EP2C15AF484I8	ADCRECV,AES128_FAST,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,RS_ENCODE,UART_16750,UCRC_PAR,SC_CORPROC
EP2C20F484C8	ADCRECV,AES128_FAST,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,RS_ENCODE,UART_16750,UCRC_PAR,SC_CORPROC
EP3C16F484C6	ADCRECV,AES128_FAST,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,RS_ENCODE,UART_16750,UCRC_PAR,SC_CORPROC
EP3C16F484C7	ADCRECV,AES128_FAST,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,RS_ENCODE,UART_16750,UCRC_PAR,SC_CORPROC
EP3C16F484C8	ADCRECV,AES128_FAST,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,RS_ENCODE,UART_16750,UCRC_PAR,SC_CORPROC
EP3C16U484C7	ADCRECV,AES128_FAST,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,RS_ENCODE,UART_16750,UCRC_PAR,SC_CORPROC
EP3C16U484C8	ADCRECV,AES128_FAST,AES_DEC,AES_ENC,SIN,SPI_BOOT,T400_CORE,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,FPU,I2CSLAVETOP,RS_DEC,RS_ENCODE,UART_16750,UCRC_PAR,SC_CORPROC

Tecnología	Circuitos_implementados
XC3S50-4CP132I	ADCRECV,SIN,SPI_BOOT,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,I2CSLAVETOP,UART_16750,UCRC_PAR,SC_CORPROC
XC3S50-4CP132C	ADCRECV,SIN,SPI_BOOT,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,I2CSLAVETOP,UART_16750,UCRC_PAR,SC_CORPROC
XC3S50-4VQ100I	ADCRECV,SIN,SPI_BOOT,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,I2CSLAVETOP,UART_16750,UCRC_PAR,SC_CORPROC
XC3S50-4VQ100C	ADCRECV,SIN,SPI_BOOT,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,I2CSLAVETOP,UART_16750,UCRC_PAR,SC_CORPROC
XC3S50-5CP132C	ADCRECV,SIN,SPI_BOOT,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,I2CSLAVETOP,UART_16750,UCRC_PAR,SC_CORPROC
XC3S50-5VQ100C	ADCRECV,SIN,SPI_BOOT,ANALYTIC_FILTER_H_A1,DDS_SYNTHESIZER,I2CSLAVETOP,UART_16750,UCRC_PAR,SC_CORPROC

Tabla 87. Circuitos implementados para algunas de las tecnologías revisadas (Tabla *Circuitos_Implementados*)

7.6. Puntuaciones globales y finales

La puntuación global, PG, es la calificación del comportamiento general de una FPGA ante un circuito. Para su cálculo se emplea la media geómetrica de las Pk' ya obtenidas para ese circuito.

La puntuación final, PF, informa del comportamiento de cada modelo de FPGA ante el PdC. Atendiendo a este valor será como finalmente se establezca la clasificación de las FPGAs analizadas.

Es posible conocer PG y PF de diferentes grupos de FPGAs:

- PG y PF del conjunto de FPGAs que alojó los 16 circuitos que forman el PdC.
- PG y PF de todas las FPGAs considerando solo los 7 circuitos implementados en común. La clasificación final de la muestra de FPGAs analizadas se construirá utilizando los resultados de este caso.
- PG y PF por sector de mercado
- PG y PF de un conjunto de FPGAs que se ajustan a las características elegidas en el interfaz de **AsistenteFPGA.**
- PG y PF de un conjunto de FPGAs filtrado según criterios impuestos por el usuario quien también decide qué circuitos participarán en la puntuación.

La BD dispone de dos tablas para almacenar los resultados obtenidos para PG y PF de cualquiera de los grupos citados. *Tabla_seleccion_FPGA* para guardar PG y *Tabla_seleccion_FPGA_fp* para PF.

A continuación se recogen los resultados obtenidos para todas los grupos posibles, en los casos en que la cantidad de registros sea excesivamente grande, se mostrará solo una parte de la tabla.

7.6.1. Todos los circuitos

La aplicación **AsistenteFPGA** permite extraer esta información si no se modifica ninguno de los criterios de búsqueda que aparecen en la pantalla inicial. De este modo, el subconjunto de FPGAs seleccionado coincide con el total existente y la puntuación se obtiene para la opción marcada por defecto en el campo Puntuaciones: “Incluir FPGAs con todos los circuitos”.

Al término de esta búsqueda la tabla *tabla_selección_fpga* guarda un total de 144 registros que recogen la PG obtenida por un total de 9 modelos de FPGAs frente a los 16 circuitos del PdC, estos últimos aparecen listados en la pantalla de presentación junto a la tabla principal.

Como ya se ha visto en apartados anteriores, el circuito denominado AES128_fast, requiere de un gran número de terminales de usuario, lo que limita su implementación a solo 9 tecnologias del conjunto. Por esta razón, la tabla *tabla_selección_fpga_fp* queda reducida a 9 modelos de los que se muestra su PG ante cada circuito del PdC, Tabla 88.

La tabla se presenta ordenada por circuitos informando de la tecnología que consiguió la mayor puntuación global en cada uno. Para todos los casos, el modelo más barato es el que alcanza la mayor puntuación, pero el aumento del precio no rige estrictamente el descenso de calificación.

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
EP3C16F484C8	ADCRECV	1,8991	1,5810	1,5631	1,5631	1,0778	1,5122	33,2
EP3C16U484C8	ADCRECV	1,7599	1,4240	1,4110	1,4110	1,0647	1,3965	36,5
EP2C15AF484C8	ADCRECV	1,6445	1,3264	1,3165	1,3165	1,0496	1,3174	38,9
EP3C16F484C7	ADCRECV	1,6666	1,2895	1,2806	1,2806	1,0534	1,3000	39,9
EP3C16U484C7	ADCRECV	1,5467	1,1579	1,1530	1,1530	1,0424	1,1994	43,9
EP2C15AF484C7	ADCRECV	1,6410	1,0789	1,0767	1,0767	1,0301	1,1615	46,7
EP2C15AF484I8	ADCRECV	1,4278	1,0789	1,0767	1,0767	1,0301	1,1297	46,7
EP2C20F484C8	ADCRECV	1,4155	1,0744	1,0716	1,0716	1,0296	1,1245	46,9
EP3C16F484C6	ADCRECV	1,5905	1,0000	1,0000	1,0000	1,0292	1,1036	49,9
EP3C16F484C8	AES_DEC	2,0192	5,1716	2,8354	2,0514	1,8956	2,5837	33,2
EP3C16U484C8	AES_DEC	1,8110	4,7808	2,4423	1,7674	1,7815	2,3156	36,5
EP3C16F484C7	AES_DEC	1,9345	4,4456	2,1052	1,5239	1,6836	2,1548	39,9
EP2C15AF484C8	AES_DEC	1,2871	4,4963	2,1640	1,5663	1,6256	1,9986	38,9
EP3C16U484C7	AES_DEC	1,7325	4,1179	1,7756	1,2857	1,5879	1,9166	43,9
EP3C16F484C6	AES_DEC	1,7408	3,7251	1,3801	1,0000	1,4731	1,6750	49,9
EP2C15AF484C7	AES_DEC	1,2644	3,8872	1,5498	1,1226	1,4598	1,6568	46,7
EP2C15AF484I8	AES_DEC	1,0248	3,8872	1,5498	1,1226	1,4598	1,5886	46,7
EP2C20F484C8	AES_DEC	1,0000	4,0025	1,5368	1,1132	1,4563	1,5840	46,9
EP3C16F484C8	AES_ENC	2,5670	4,4438	2,8354	2,0514	1,8936	2,6292	33,2
EP3C16U484C8	AES_ENC	2,2529	4,0040	2,4423	1,7674	1,7794	2,3341	36,5
EP3C16F484C7	AES_ENC	2,2204	3,6269	2,1052	1,5239	1,6816	2,1261	39,9
EP2C15AF484C8	AES_ENC	1,5830	3,6911	2,1640	1,5663	1,6367	2,0052	38,9
EP3C16U484C7	AES_ENC	1,9354	3,2581	1,7756	1,2857	1,5858	1,8694	43,9
EP3C16F484C6	AES_ENC	1,7693	2,8157	1,3801	1,0000	1,4710	1,5885	49,9
EP2C15AF484C7	AES_ENC	1,2427	3,0040	1,5498	1,1226	1,4687	1,5700	46,7
EP2C15AF484I8	AES_ENC	1,1670	3,0042	1,5498	1,1226	1,4687	1,5504	46,7
EP2C20F484C8	AES_ENC	1,0000	3,1115	1,5368	1,1132	1,4652	1,5080	46,9
EP3C16F484C8	AES128_FAST	8,8694	10,0000	10,0000	10,0000	10,000	9,7629	33,2
EP3C16U484C8	AES128_FAST	7,0131	7,7830	7,5687	7,5687	8,0380	7,5866	36,5
EP3C16F484C7	AES128_FAST	10,0000	5,8824	5,4843	5,4843	6,3561	6,4595	39,9
EP2C15AF484C8	AES128_FAST	3,7842	4,1122	6,0595	6,0595	3,6119	4,6018	38,9
EP3C16U484C7	AES128_FAST	8,0262	4,0233	3,4454	3,4454	4,7108	4,4806	43,9
EP3C16F484C6	AES128_FAST	7,6341	1,7935	1,0000	1,0000	2,7375	2,0643	49,9
EP2C20F484C8	AES128_FAST	1,0000	7,9451	2,1445	2,1445	1,0000	2,0538	46,9
EP2C15AF484C7	AES128_FAST	4,1943	1,0000	2,2260	2,2260	1,0544	1,8541	46,7
EP2C15AF484I8	AES128_FAST	1,2042	1,0000	2,2260	2,2260	1,0544	1,4446	46,7
EP3C16F484C8	ANALYTIC_FILTER_H_A1	2,9674	2,2661	2,0368	2,7419	1,7978	2,3221	33,2
EP3C16U484C8	ANALYTIC_FILTER_H_A1	2,6350	1,9241	1,7567	2,3301	1,6945	2,0381	36,5
EP3C16F484C7	ANALYTIC_FILTER_H_A1	2,6004	1,6308	1,5166	1,9771	1,6059	1,8282	39,9

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
EP2C15AF484C8	ANALYTIC_FILTER_H_A1	2,0574	1,7087	1,5823	2,0738	1,5543	1,7812	38,9
EP3C16U484C7	ANALYTIC_FILTER_H_A1	2,2989	1,3440	1,2817	1,6318	1,5192	1,5791	43,9
EP2C15AF484C7	ANALYTIC_FILTER_H_A1	1,8436	1,1699	1,1408	1,4247	1,4041	1,3754	46,7
EP2C20F484C8	ANALYTIC_FILTER_H_A1	1,6891	1,1684	1,1314	1,4109	1,4009	1,3457	46,9
EP3C16F484C6	ANALYTIC_FILTER_H_A1	2,4859	1,0000	1,0000	1,2177	1,4153	1,3377	49,9
EP2C15AF484I8	ANALYTIC_FILTER_H_A1	1,5954	1,1699	1,1408	1,4247	1,4041	1,3362	46,7
EP3C16F484C8	DDS_SYNTHESIZER	1,6713	1,6127	1,5631	1,5631	1,2086	1,5141	33,2
EP3C16U484C8	DDS_SYNTHESIZER	1,5036	1,4472	1,4110	1,4110	1,1816	1,3862	36,5
EP3C16F484C7	DDS_SYNTHESIZER	1,5739	1,3053	1,2806	1,2806	1,1585	1,3130	39,9
EP2C15AF484C8	DDS_SYNTHESIZER	1,2724	1,3438	1,3165	1,3165	1,1482	1,2775	38,9
EP3C16U484C7	DDS_SYNTHESIZER	1,4139	1,1665	1,1530	1,1530	1,1359	1,2002	43,9
EP2C15AF484C7	DDS_SYNTHESIZER	1,2342	1,0829	1,0767	1,0767	1,1085	1,1142	46,7
EP3C16F484C6	DDS_SYNTHESIZER	1,4552	1,0000	1,0000	1,0000	1,1088	1,1004	49,9
EP2C15AF484I8	DDS_SYNTHESIZER	1,0294	1,0829	1,0767	1,0767	1,1085	1,0745	46,7
EP2C20F484C8	DDS_SYNTHESIZER	1,0234	1,0794	1,0716	1,0716	1,1076	1,0704	46,9
EP3C16F484C8	FPU	4,6315	8,8072	2,7480	3,0656	2,9007	3,9785	33,2
EP3C16U484C8	FPU	4,2399	8,0822	2,2758	2,6140	2,6585	3,5220	36,5
EP3C16F484C7	FPU	4,3432	7,4666	1,8709	2,2269	2,4509	3,1916	39,9
EP2C15AF484C8	FPU	3,3592	7,4572	1,9826	2,3338	2,2349	3,0386	38,9
EP3C16U484C7	FPU	3,9748	6,8581	1,4749	1,8483	2,2478	2,7833	43,9
EP2C15AF484C7	FPU	3,1406	6,3450	1,2381	1,6219	1,9171	2,3822	46,7
EP2C20F484C8	FPU	2,8172	6,8476	1,2223	1,6067	1,8915	2,3499	46,9
EP3C16F484C6	FPU	4,1039	6,1235	1,0000	1,3942	2,0043	2,3404	49,9
EP2C15AF484I8	FPU	2,8483	6,3433	1,2381	1,6219	1,8987	2,3315	46,7
EP3C16F484C8	I2CSLAVETOP	2,3846	1,7468	1,5631	1,5631	1,1969	1,6487	33,2
EP3C16U484C8	I2CSLAVETOP	2,1684	1,5450	1,4110	1,4110	1,1721	1,5087	36,5
EP3C16F484C7	I2CSLAVETOP	2,1581	1,3720	1,2806	1,2806	1,1508	1,4107	39,9
EP2C15AF484C8	I2CSLAVETOP	1,6764	1,4179	1,3165	1,3165	1,1433	1,3634	38,9
EP3C16U484C7	I2CSLAVETOP	1,9608	1,2028	1,1530	1,1530	1,1300	1,2879	43,9
EP2C15AF484C7	I2CSLAVETOP	1,7455	1,1001	1,0767	1,0767	1,1064	1,1976	46,7
EP3C16F484C6	I2CSLAVETOP	2,1539	1,0000	1,0000	1,0000	1,1051	1,1894	49,9
EP2C20F484C8	I2CSLAVETOP	1,5186	1,0990	1,0716	1,0716	1,1056	1,1621	46,9
EP2C15AF484I8	I2CSLAVETOP	1,3953	1,1001	1,0767	1,0767	1,1064	1,1451	46,7
EP3C16F484C8	RS_DEC	3,3490	7,6333	3,9645	2,6118	2,5137	3,6696	33,2
EP3C16U484C8	RS_DEC	2,8819	7,0142	3,6148	2,1837	2,3208	3,2637	36,5
EP3C16F484C7	RS_DEC	2,7000	6,4872	3,3150	1,8167	2,1554	2,9604	39,9

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
EP2C15AF484C8	RS_DEC	1,8881	6,4427	3,1725	1,6423	2,0417	2,6447	38,9
EP3C16U484C7	RS_DEC	2,2884	5,9677	3,0218	1,4577	1,9936	2,6049	43,9
EP3C16F484C6	RS_DEC	2,3867	5,3443	2,6701	1,0272	1,7996	2,2898	49,9
EP2C15AF484C7	RS_DEC	1,7220	5,4999	2,6588	1,0134	1,7641	2,1413	46,7
EP2C15AF484I8	RS_DEC	1,2692	5,4979	2,6588	1,0134	1,7641	2,0144	46,7
EP2C20F484C8	RS_DEC	1,1010	6,0149	2,6479	1,0000	1,7582	1,9852	46,9
EP3C16F484C8	RS_ENCODE	9,1219	6,0392	4,5408	4,5408	2,0397	4,7096	33,2
EP3C16U484C8	RS_ENCODE	7,7677	4,9024	3,5843	3,5843	1,8064	3,8839	36,5
EP3C16F484C7	RS_ENCODE	9,0526	3,9278	2,7642	2,7642	1,6064	3,3728	39,9
EP2C15AF484C8	RS_ENCODE	5,0469	4,1501	2,9905	2,9905	1,3169	3,0090	38,9
EP3C16U484C7	RS_ENCODE	7,6942	2,9746	1,9621	1,9621	1,4108	2,6236	43,9
EP2C15AF484C7	RS_ENCODE	5,0464	2,3656	1,4823	1,4823	1,0066	1,9246	46,7
EP2C20F484C8	RS_ENCODE	3,5122	2,4701	1,4503	1,4503	1,0000	1,7875	46,9
EP2C15AF484I8	RS_ENCODE	3,2259	2,3656	1,4823	1,4823	1,0066	1,7598	46,7
EP3C16F484C6	RS_ENCODE	7,6546	1,8312	1,0000	1,0000	1,1761	1,7515	49,9
EP3C16F484C8	SC_CORPROC	1,9129	2,3208	1,5631	1,5631	1,1935	1,6689	33,2
EP3C16U484C8	SC_CORPROC	1,7267	1,9671	1,4110	1,4110	1,1650	1,5111	36,5
EP3C16F484C7	SC_CORPROC	1,8419	1,6638	1,2806	1,2806	1,1471	1,4196	39,9
EP2C15AF484C8	SC_CORPROC	1,4317	1,7314	1,3165	1,3165	1,1318	1,3721	38,9
EP3C16U484C7	SC_CORPROC	1,6607	1,3672	1,1530	1,1530	1,1232	1,2766	43,9
EP2C15AF484C7	SC_CORPROC	1,3313	1,1764	1,0767	1,0767	1,0959	1,1475	46,7
EP2C15AF484I8	SC_CORPROC	1,1681	1,1764	1,0767	1,0767	1,0959	1,1179	46,7
EP3C16F484C6	SC_CORPROC	1,5507	1,0114	1,0000	1,0000	1,1010	1,1155	49,9
EP2C20F484C8	SC_CORPROC	1,1200	1,2145	1,0716	1,0716	1,0952	1,1133	46,9
EP3C16F484C8	SIN	1,8326	1,5762	1,5631	1,5631	1,1915	1,5309	33,2
EP3C16U484C8	SIN	1,6614	1,4205	1,4110	1,4110	1,1678	1,4056	36,5
EP3C16F484C7	SIN	1,7195	1,2871	1,2806	1,2806	1,1474	1,3302	39,9
EP2C15AF484C8	SIN	1,3547	1,3236	1,3165	1,3165	1,1351	1,2868	38,9
EP3C16U484C7	SIN	1,5573	1,1565	1,1530	1,1530	1,1275	1,2197	43,9
EP2C15AF484C7	SIN	1,3276	1,0783	1,0767	1,0767	1,1007	1,1281	46,7
EP3C16F484C6	SIN	1,5128	1,0000	1,0000	1,0000	1,1037	1,1080	49,9
EP2C20F484C8	SIN	1,1607	1,0738	1,0716	1,0716	1,1023	1,0955	46,9
EP2C15AF484I8	SIN	1,1182	1,0783	1,0767	1,0767	1,1007	1,0900	46,7
EP3C16F484C8	SPI_BOOT	1,9231	1,6246	1,5631	1,5631	1,2100	1,5599	33,2
EP3C16U484C8	SPI_BOOT	1,7471	1,4559	1,4110	1,4110	1,1835	1,4306	36,5
EP3C16F484C7	SPI_BOOT	1,8835	1,3112	1,2806	1,2806	1,1608	1,3628	39,9
EP2C15AF484C8	SPI_BOOT	1,4364	1,3504	1,3165	1,3165	1,1530	1,3113	38,9
EP3C16U484C7	SPI_BOOT	1,7097	1,1697	1,1530	1,1530	1,1386	1,2480	43,9
EP2C15AF484C7	SPI_BOOT	1,4023	1,0845	1,0767	1,0767	1,1136	1,1445	46,7
EP3C16F484C6	SPI_BOOT	1,6695	1,0000	1,0000	1,0000	1,1120	1,1317	49,9
EP2C20F484C8	SPI_BOOT	1,2178	1,0810	1,0716	1,0716	1,1127	1,1096	46,9
EP2C15AF484I8	SPI_BOOT	1,1925	1,0845	1,0767	1,0767	1,1136	1,1080	46,7

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
EP3C16F484C8	T400_CORE	2,7958	3,1529	1,7933	1,7933	1,3611	2,0763	33,2
EP3C16U484C8	T400_CORE	2,4956	2,6118	1,5790	1,5790	1,3147	1,8448	36,5
EP3C16F484C7	T400_CORE	2,7024	2,1478	1,3952	1,3952	1,2750	1,7049	39,9
EP2C15AF484C8	T400_CORE	1,8880	2,2610	1,4459	1,4459	1,2396	1,6173	38,9
EP3C16U484C7	T400_CORE	2,4083	1,6940	1,2155	1,2155	1,2361	1,4943	43,9
EP2C15AF484C7	T400_CORE	1,8771	1,4103	1,1081	1,1081	1,1743	1,3072	46,7
EP3C16F484C6	T400_CORE	2,3391	1,1497	1,0000	1,0000	1,1895	1,2618	49,9
EP2C20F484C8	T400_CORE	1,5085	1,4375	1,1009	1,1009	1,1729	1,2525	46,9
EP2C15AF484I8	T400_CORE	1,4850	1,4103	1,1081	1,1081	1,1743	1,2473	46,7
EP3C16F484C8	UART_16750	2,1666	2,3942	1,9162	1,5616	1,2232	1,8017	33,2
EP3C16U484C8	UART_16750	1,9672	2,0432	1,6741	1,4099	1,1955	1,6253	36,5
EP3C16F484C7	UART_16750	2,1596	1,7426	1,4666	1,2798	1,1718	1,5261	39,9
EP2C15AF484C8	UART_16750	1,4681	1,8186	1,5176	1,3118	1,1606	1,4390	38,9
EP3C16U484C7	UART_16750	1,9593	1,4483	1,2636	1,1526	1,1485	1,3655	43,9
EP2C15AF484C7	UART_16750	1,6923	1,2663	1,1370	1,0733	1,1199	1,2397	46,7
EP2C20F484C8	UART_16750	1,2634	1,2774	1,1289	1,0682	1,1190	1,1684	46,9
EP3C16F484C6	UART_16750	1,7307	1,0952	1,0201	1,0000	1,1207	1,1673	49,9
EP2C15AF484I8	UART_16750	1,2165	1,2665	1,1370	1,0733	1,1199	1,1606	46,7
EP3C16F484C8	UCRC_PAR	2,6699	1,5721	1,5631	1,5631	1,1933	1,6502	33,2
EP3C16U484C8	UCRC_PAR	2,4287	1,4176	1,4110	1,4110	1,1688	1,5161	36,5
EP3C16F484C7	UCRC_PAR	2,5078	1,2851	1,2806	1,2806	1,1479	1,4341	39,9
EP2C15AF484C8	UCRC_PAR	1,9641	1,3215	1,3165	1,3165	1,1407	1,3869	38,9
EP3C16U484C7	UCRC_PAR	2,2795	1,1555	1,1530	1,1530	1,1274	1,3160	43,9
EP3C16F484C6	UCRC_PAR	2,3109	1,0000	1,0000	1,0000	1,1028	1,2057	49,9
EP2C15AF484C7	UCRC_PAR	1,7725	1,0778	1,0767	1,0767	1,1043	1,1959	46,7
EP2C15AF484I8	UCRC_PAR	1,6364	1,0778	1,0767	1,0767	1,1043	1,1769	46,7
EP2C20F484C8	UCRC_PAR	1,4743	1,0731	1,0716	1,0716	1,1035	1,1493	46,9

Tabla 88. Puntuación global de todas las tecnologías que completaron el PdC (9 Tecnologías, 16 circuitos y 144 registros en total)

En el caso de la PF, Tabla 89, el precio sigue siendo determinante a la hora de establecer un orden aunque no en el 100% de los casos.

Tecnología	P_Final	Precio	Pines	U_logicas	Memoria	Mult	Encapsulado	Tmax	Tmin
EP3C16F484C8	2,3689	33,2	346	15408	516096	56	484FBGA	85	0
EP3C16U484C8	2,1067	36,5	346	15408	516096	56	484UBGA	85	0
EP3C16F484C7	1,9388	39,9	346	15408	516096	56	484FBGA	85	0

Tecnología	P_Final	Precio	Pines	U_logicas	Memoria	Mult	Encapsulado	Tmax	Tmin
EP2C15AF484C8	1,8129	38,9	315	14448	239616	26	484FBGA	85	0
EP3C16U484C7	1,6955	43,9	346	15408	516096	56	484UBGA	85	0
EP2C15AF484C7	1,4252	46,7	315	14448	239616	26	484FBGA	85	0
EP3C16F484C6	1,4112	49,9	346	15408	516096	56	484FBGA	85	0
EP2C20F484C8	1,3811	46,9	315	18752	239616	26	484FBGA	85	0
EP2C15AF484I8	1,3527	46,7	315	14448	239616	26	484FBGA	100	-40

Tabla 89. Puntuación final para todas las FPGA que completaron el PdC

7.6.2. Todas las FPGAs

Para conseguir puntuar en igualdad todas las FPGA de la muestra se hace necesario eliminar aquellos circuitos del PdC que no se ajustaron a todas ellas. Haciendo uso de la herramienta **AsistenteFPGA** es posible filtrar los resultados y limitarlos a los 7 circuitos implementados con éxito en todas las tecnologías. Al igual que en el apartado anterior, los criterios de búsqueda por características constructivas se mantienen en su opción por defecto y la única modificación se practica en el campo Puntuaciones donde se selecciona “Todas las FPGAS con circuitos comunes”.

El fruto de esta búsqueda queda almacenado en la misma tabla denominada *Tabla_seleccion_FPGA* sobreescribiendo los resultados obtenidos anteriormente. El conjunto de registros suma un total de 1.722, en la Tabla 90 se presentan los cinco primeros registros de cada uno de los 7 núcleos, tras ordenar la tabla por circuitos y en orden creciente de puntuación global.

El precio sigue actúando como principal peso en la clasificación, razón por la cual las tecnologías de Xilinx, se ven claramente beneficiadas al ofrecer los precios más bajos.

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
XC3S50A-4VQ100C	DDS_SYNTHESIZER	9,8085	10,0000	10,0000	10,0000	10,0000	9,9614	5,52
XC3S50A-5VQ100C	DDS_SYNTHESIZER	10,0000	8,3280	8,3441	8,3441	8,5126	8,6830	6,6
XC3S50A-4VQ100I	DDS_SYNTHESIZER	8,4889	8,6504	8,6635	8,6635	8,7995	8,6526	6,36
XC3S50A-4TQ144C	DDS_SYNTHESIZER	7,1701	7,3631	7,3885	7,3885	7,6543	7,3913	7,44
XC3S50A-5TQ144C	DDS_SYNTHESIZER	7,4639	6,0911	6,1288	6,1288	6,5228	6,4472	8,94
XC3S50A-4VQ100C	I2CSLAVETOP	9,5600	10,0000	10,0000	10,0000	10,0000	9,9104	5,52
XC3S50A-5VQ100C	I2CSLAVETOP	10,0000	8,1872	8,3441	8,3441	8,5146	8,6539	6,6
XC3S50A-4VQ100I	I2CSLAVETOP	8,2965	8,6152	8,6635	8,6635	8,8011	8,6063	6,36
XC3S50A-4TQ144C	I2CSLAVETOP	6,7237	7,2942	7,3885	7,3885	7,6574	7,2838	7,44
XC3S50A-4TQ144I	I2CSLAVETOP	5,8294	6,2606	6,3910	6,3910	6,7626	6,3197	8,58
XC3S50A-4VQ100C	SC_CORPROC	10,0000	10,0000	10,0000	10,0000	10,0000	10,0000	5,52
XC3S50A-4VQ100I	SC_CORPROC	8,6599	8,4690	8,6635	8,6635	8,8004	8,6506	6,36
XC3S50A-5VQ100C	SC_CORPROC	9,5140	7,7130	8,3441	8,3441	8,5137	8,4663	6,6
XC3S50A-4TQ144C	SC_CORPROC	7,8987	7,0086	7,3885	7,3885	7,6560	7,4621	7,44
XC3S50A-4TQ144I	SC_CORPROC	6,8298	5,8659	6,3910	6,3910	6,7606	6,4383	8,58
XC3S50A-4FT256C	SC_CORPROC	6,4242	5,7127	6,2572	6,2572	6,6405	6,2506	8,76
XC3S50A-5TQ144C	SC_CORPROC	7,2995	5,2775	6,1288	6,1288	6,5253	6,2376	8,94
XC3S50AN-4TQG144C	SC_CORPROC	6,5488	5,5656	6,1288	6,1288	5,4737	5,9558	8,94
XC3S50A-4FT256I	SC_CORPROC	5,5637	4,7561	5,4222	5,4222	5,8910	5,3980	10,08
XC3S50A-5FT256C	SC_CORPROC	6,3044	4,2237	5,1703	5,1703	5,6649	5,2615	10,56
XC3S50A-4VQ100C	SIN	10,0000	10,0000	10,0000	10,0000	10,0000	10,0000	5,52
XC3S50A-4VQ100I	SIN	8,6712	8,6600	8,6635	8,6635	8,8020	8,6919	6,36
XC3S50A-5VQ100C	SIN	9,6339	8,3398	8,3441	8,3441	8,5157	8,6216	6,6
XC3S50A-4TQ144C	SIN	7,1833	7,3818	7,3885	7,3885	7,6591	7,3987	7,44
XC3S50A-4TQ144I	SIN	6,2208	6,3817	6,3910	6,3910	6,7650	6,4274	8,58
XC3S50A-4VQ100C	SPI_BOOT	10,0000	10,0000	10,0000	10,0000	10,0000	10,0000	5,52
XC3S50A-4VQ100I	SPI_BOOT	8,6761	8,6473	8,6635	8,6635	8,8004	8,6900	6,36
XC3S50A-5VQ100C	SPI_BOOT	9,4410	8,2838	8,3441	8,3441	8,5137	8,5748	6,6
XC3S50A-4TQ144C	SPI_BOOT	7,7512	7,3570	7,3885	7,3885	7,6560	7,5065	7,44
XC3S50A-4TQ144I	SPI_BOOT	6,7182	6,3474	6,3910	6,3910	6,7606	6,5192	8,58
XC3S50A-4VQ100C	UART_16750	10,0000	10,0000	10,0000	10,0000	10,0000	10,0000	5,52
XC3S50A-4VQ100I	UART_16750	8,6742	8,4828	8,5787	8,6639	8,8003	8,6393	6,36
XC3S50A-5VQ100C	UART_16750	9,4279	8,3497	8,2391	8,3446	8,5137	8,5644	6,6
XC3S50A-4TQ144C	UART_16750	7,4747	7,0356	7,2229	7,3893	7,6559	7,3526	7,44
XC3S50A-4TQ144I	UART_16750	6,4765	5,9032	6,1620	6,3920	6,7605	6,3322	8,58
XC3S50A-4VQ100C	UCRC_PAR	10,0000	10,0000	10,0000	10,0000	10,0000	10,0000	5,52

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
XC3S50A-4VQ100I	UCRC_PAR	8,6795	8,6611	8,6635	8,6635	8,8011	8,6936	6,36
XC3S50A-5VQ100C	UCRC_PAR	9,7533	8,3411	8,3441	8,3441	8,5147	8,6429	6,6
XC3S50A-4TQ144C	UCRC_PAR	7,0274	7,3839	7,3885	7,3885	7,6575	7,3664	7,44
XC3S50A-4TQ144I	UCRC_PAR	6,0939	6,3845	6,3910	6,3910	6,7627	6,4011	8,58

Tabla 90.Puntuación global de todas las FPGAs ante 7 circuitos (Muestra de *tabla_seleccion_FPGA)*

Para la PF el total de registros coincide con el total de FPGAs, 246, razón por la que la Tabla 91 solo muestra 40 elementos, los 40 con mejor PF.

Tecnología	P_Final	Precio	Pines	U_log.	Memo.	Mult	Encaps	Tmax	Tmin
XC3S50A-4VQ100C	9,9816	5,52	68	1408	55296	3	100VTQFP	85	0
XC3S50A-4VQ100I	8,6605	6,36	68	1408	55296	3	100VTQFP	100	-40
XC3S50A-5VQ100C	8,6007	6,6	68	1408	55296	3	100VTQFP	85	0
XC3S50A-4TQ144C	7,3942	7,44	108	1408	55296	3	144TQFP EP	85	0
XC3S50A-4TQ144I	6,4073	8,58	108	1408	55296	3	144TQFP EP	100	-40
XC3S50A-5TQ144C	6,3410	8,94	108	1408	55296	3	144TQFP EP	85	0
XC3S50A-4FT256C	6,2845	8,76	144	1408	55296	3	256FTBGA	85	0
XC3S50AN-4TQG144C	5,9610	8,94	108	1408	55296	3	144TQFP EP	85	0
XC3S50A-4FT256I	5,4560	10,08	144	1408	55296	3	256FTBGA	100	-40
XC3S50A-5FT256C	5,3726	10,56	144	1408	55296	3	256FTBGA	85	0
XC3S50AN-4TQG144I	5,1619	10,32	108	1408	55296	3	144TQFP EP	100	-40
XC3S50-4CP132C	5,1407	9,3	89	1536	73728	50	132CSBGA	85	0
XC3S50-4TQ144C	5,1309	9,3	97	1536	73728	50	144TQFP	85	0
XC3S50AN-5TQG144C	5,1228	10,74	108	1408	55296	3	144TQFP EP	85	0
XC3S100E-4VQ100C	4,9981	9,48	66	1920	73728	4	100VTQFP	85	0
XC3S50-5VQ100C	4,8572	10,1	63	1536	73728	50	100VTQFP	85	0

Tecnología	P_Final	Precio	Pines	U_log.	Memo.	Mult	Encaps	Tmax	Tmin
XC3S50-4VQ100C	4,7219	10,1	63	1536	73728	50	100VTQFP	85	0
XC3S50-4VQ100I	4,7219	10,1	63	1536	73728	50	100VTQFP	100	-40
XC3S50-5CP132C	4,6024	10,69997	89	1536	73728	50	132CSBGA	85	0
XC3S50-5TQ144C	4,5965	10,7	97	1536	73728	50	144TQFP	85	0
XC3S100E-5VQ100C	4,5187	10,92	66	1920	73728	4	100VTQFP	85	0
XC3S200A-4VQ100C	4,5105	11,6	68	3584	294912	16	100VTQFP	85	0
XC3S50-4CP132I	4,4850	10,69997	89	1536	73728	50	132CSBGA	100	-40
XC3S50-4TQ144I	4,4766	10,7	97	1536	73728	50	144TQFP	100	-40
XC3S100E-4VQ100I	4,3628	10,92	66	1920	73728	4	100VTQFP	100	-40
XC3S100E-4CP132C	4,3103	11,1	83	1920	73728	4	132CSBGA	85	0
XC3S100E-4CP132I	4,3103	11,1	83	1920	73728	4	132CSBGA	100	-40
XC3S200-4VQ100C	4,1433	11,7	63	3840	221184	12	100VTQFP	85	0
XC3S200A-4VQ100I	4,0850	12,84	68	3584	294912	16	100VTQFP	100	-40
XC3S200A-5VQ100C	4,0475	13,44	68	3584	294912	16	100VTQFP	85	0
EP2C5T144C8	4,0296	12,8	89	4608	119808	13	144TQFP	85	0
XA3S100E-4VQG100I	3,9659	12,06	66	1920	73728	4	100VTQFP	100	-40
XC3S50-4PQ208C	3,9536	12,1	124	1536	73728	50	208PQFP	85	0
EP3C5E144C8	3,9321	12,8	94	5136	423936	23	144-EQFP	85	0
XC3S200A-4FT256C	3,8949	13,44	195	3584	294912	16	256FTBGA	85	0
XC3S100E-5TQ144C	3,8849	12,78	108	1920	73728	4	144TQFPEP	85	0
XC3S100E-5CP132C	3,8802	12,78	83	1920	73728	4	132CSBGA	85	0
XC3S100E-4TQ144C	3,7804	12,78	108	1920	73728	4	144TQFPEP	85	0
XC3S100E-4TQ144I	3,7804	12,78	108	1920	73728	4	144TQFPEP	100	-40

Tabla 91. Clasificación de FPGAs, las 40 primeras.

A continuación, Tabla 92, se adjunta al completo la clasificación final de las 246 FPGAs según los 7 circuitos válidos para todas. La ordenación está basada en las PF obtenidas por cada una al término de este trabajo y se constrasta con el orden que establecería el precio.

La última columna de la tabla indica las posiciones que ganan, positivo, o pierden, negativo, cada modelo respecto a la clasificación por precios.

Tecnología	P_Final	Orden por PF	Precio	Orden por precio	Arriba/abajo
XC3S50A-4VQ100C	9,9816	1	5,52	1	0
XC3S50A-4VQ100I	8,6605	2	6,36	2	0
XC3S50A-5VQ100C	8,6007	3	6,6	3	0
XC3S50A-4TQ144C	7,3942	4	7,44	4	0
XC3S50A-4TQ144I	6,4073	5	8,58	5	0
XC3S50A-5TQ144C	6,3410	6	8,94	7	1
XC3S50A-4FT256C	6,2845	7	8,76	6	-1
XC3S50AN-4TQG144C	5,9610	8	8,94	8	0
XC3S50A-4FT256I	5,4560	9	10,08	12	3
XC3S50A-5FT256C	5,3726	10	10,56	17	7
XC3S50AN-4TQG144I	5,1619	11	10,32	16	5
XC3S50-4CP132C	5,1407	12	9,3	9	-3
XC3S50-4TQ144C	5,1309	13	9,3	10	-3
XC3S50AN-5TQG144C	5,1228	14	10,74	22	8
XC3S100E-4VQ100C	4,9981	15	9,48	11	-4
XC3S50-5VQ100C	4,8572	16	10,1	13	-3
XC3S50-4VQ100I	4,7219	18	10,1	15	-3
XC3S50-4VQ100C	4,7219	17	10,1	14	-3
XC3S50-5CP132C	4,6024	19	10,69997	18	-1
XC3S50-5TQ144C	4,5965	20	10,7	20	0
XC3S100E-5VQ100C	4,5187	21	10,92	23	2
XC3S200A-4VQ100C	4,5105	22	11,6	27	5
XC3S50-4CP132I	4,4850	23	10,69997	19	-4
XC3S50-4TQ144I	4,4766	24	10,7	21	-3
XC3S100E-4VQ100I	4,3628	25	10,92	24	-1
XC3S100E-4CP132I	4,3103	27	11,1	26	-1
XC3S100E-4CP132C	4,3103	26	11,1	25	-1
XC3S200-4VQ100C	4,1433	28	11,7	28	0
XC3S200A-4VQ100I	4,0850	29	12,84	37	8
XC3S200A-5VQ100C	4,0475	30	13,44	39	9
EP2C5T144C8	4,0296	31	12,8	35	4
XA3S100E-4VQG100I	3,9659	32	12,06	29	-3
XC3S50-4PQ208C	3,9536	33	12,1	30	-3
EP3C5E144C8	3,9321	34	12,8	36	2
XC3S200A-4FT256C	3,8949	35	13,44	40	5

Tecnología	P_Final	Orden por PF	Precio	Orden por precio	Arriba/abajo
XC3S100E-5TQ144C	3,8849	36	12,78	31	-5
XC3S100E-5CP132C	3,8802	37	12,78	32	-5
XC3S100E-4TQ144I	3,7804	39	12,78	34	-5
XC3S100E-4TQ144C	3,7804	38	12,78	33	-5
XC3S200-5VQ100C	3,7260	40	13,45	41	1
EP2C5Q208C8	3,7076	41	13,9	44	3
XC3S250E-4VQ100C	3,6691	42	13,08	38	-4
XC3S200-4VQ100I	3,6319	43	13,45	42	-1
XC3S50-5PQ208C	3,5465	44	13,9	45	1
EP2C5T144C7	3,5118	45	15,3	56	11
EP2C5F256C8	3,4792	46	14,9	49	3
XA3S100E-4VQG100Q	3,4698	47	13,86	43	-4
XC3S50-4PQ208I	3,4537	48	13,9	46	-2
EP3C5F256C8	3,4438	49	14,9	50	1
XA3S100E-4TQG144I	3,4397	50	14,1	47	-3
XA3S100E-4CPG132I	3,4254	51	14,1	48	-3
EP3C5E144C7	3,4079	52	15,3	57	5
XC3S200A-4FT256I	3,4061	53	15,42	61	8
EP2C5T144I8	3,3969	54	15,3	58	4
XC3S200A-5FT256C	3,3880	55	16,14	65	10
XC3S200A-4FG320C	3,3799	56	15,78	62	6
XC3S400A-4FT256C	3,3119	57	15,84	63	6
XC3S250E-5VQ100C	3,3036	58	15,06	51	-7
XC3S200-5TQ144C	3,2844	59	15,1	53	-6
XC3S200AN-4FTG256C	3,2523	60	15,84	64	4
XC3S200-4TQ144I	3,2199	62	15,1	55	-7
XC3S200-4TQ144C	3,2199	61	15,1	54	-7
XC3S250E-4VQ100I	3,2170	63	15,06	52	-11
EP2C5Q208C7	3,1910	64	16,7	70	6
XC3S250E-4TQ144C	3,1638	65	15,3	59	-6
XC3S250E-4CP132C	3,1408	66	15,3	60	-6
EP3C5U256C8	3,1319	67	16,5	68	1
EP2C5Q208I8	3,1102	68	16,7	71	3
EP2C5F256C7	3,0135	69	17,9	76	7
XA3S100E-4TQG144Q	2,9989	70	16,26	66	-4
XA3S100E-4CPG132Q	2,9866	71	16,26	67	-4
EP3C5F256C7	2,9657	72	17,9	77	5
XC3S200A-4FG320I	2,9524	73	18,12	80	7
EP3C5M164C7	2,9485	74	17,9	78	4
XA3S250E-4VQG100I	2,9246	75	16,68	69	-6

Tecnología	P_Final	Orden por PF	Precio	Orden por precio	Arriba/abajo
EP2C5T144C6	2,9232	76	19,2	89	13
EP2C5F256I8	2,9188	77	17,9	79	2
XC3S200A-5FG320C	2,8940	78	18,96	85	7
XC3S400A-4FT256I	2,8899	79	18,24	82	3
XC3S250E-5CP132C	2,8765	80	17,58	72	-8
XC3S250E-5TQ144C	2,8720	81	17,58	73	-8
XC3S400A-5FT256C	2,8708	82	19,02	87	5
XC3S200AN-4FTG256I	2,8456	83	18,18	81	-2
XC3S200AN-5FTG256C	2,8411	84	18,96	86	2
XC3S400A-4FG320C	2,8328	85	18,66	84	-1
XC3S250E-4TQ144I	2,7791	86	17,58	74	-12
EP3C5E144I7	2,7665	87	19,2	90	3
XC3S250E-4CP132I	2,7590	88	17,58	75	-13
EP3C10E144C8	2,7187	89	19,2	91	2
EP3C5U256C7	2,7123	90	19,7	96	6
EP2C8T144C8	2,6912	91	19,2	92	1
XC3S250E-4PQ208C	2,6776	92	18,3	83	-9
XA3S250E-4VQG100Q	2,5644	93	19,2	93	0
EP2C8Q208C8	2,5401	94	20,3	101	7
XC3S400-4TQ144C	2,5331	95	19,1	88	-7
XC3S200-5PQ208C	2,5255	96	20,25	98	2
EP2C5F256C6	2,5245	97	22,4	110	13
XA3S250E-4TQG144I	2,5224	98	19,5	94	-4
XA3S250E-4CPG132I	2,5042	99	19,5	95	-4
XC3S400A-4FG320I	2,4785	100	21,42	106	6
XC3S200-4PQ208I	2,4486	102	20,25	100	-2
XC3S200-4PQ208C	2,4486	101	20,25	99	-2
XC3S250E-5PQ208C	2,4402	103	21,06	104	1
XC3S400A-4FG400C	2,4325	104	21,9	107	3
XC3S250E-4FT256C	2,4322	105	20,22	97	-8
EP3C10E144C7	2,3592	106	23,1	114	8
XC3S400A-5FG320C	2,3489	107	23,38	120	13
XC3S250E-4PQ208I	2,3477	108	21,06	105	-3
XC3S500E-4CP132C	2,3400	109	20,76	102	-7
XC3S700A-4FT256C	2,3304	110	22,5	113	3
EP2C8T144C7	2,3264	111	23,1	115	4
XC3S500E-4VQ100C	2,3256	112	20,76	103	-9
XC3S400-5TQ144C	2,2786	113	21,9885	109	-4
EP3C10F256C8	2,2690	114	23,5	121	7
EP2C8T144I8	2,2599	115	23,1	116	1

Tecnología	P_Final	Orden por PF	Precio	Orden por precio	Arriba/abajo
XC3S400-4TQ144I	2,2260	116	21,95	108	-8
EP2C8Q208C7	2,2260	117	24,3	126	9
EP2C8F256C8	2,2229	118	23,5	122	4
XA3S250E-4TQG144Q	2,2112	119	22,44	111	-8
XC3S250E-5FT256C	2,2046	120	23,28	117	-3
XA3S250E-4CPG132Q	2,1953	121	22,44	112	-9
EP2C8Q208I8	2,1426	122	24,3	127	5
XA3S250E-4PQG208I	2,1315	123	23,34	119	-4
XC3S250E-4FT256I	2,1307	124	23,28	118	-6
XC3S400A-4FG400I	2,1220	125	25,2	131	6
XC3S500E-5CP132C	2,1186	126	23,88	123	-3
XC3S400AN-4FGG400C	2,1010	127	25,14	130	3
EP3C5F256C6	2,0930	128	26,8	144	16
XC3S400A-5FG400C	2,0833	129	26,28	141	12
EP3C10U256C8	2,0788	130	25,8	138	8
EP3C5E144A7	2,0577	131	26,4	142	11
XC3S500E-4CP132I	2,0574	132	23,88	124	-8
XC3S500E-4VQ100I	2,0447	133	23,88	125	-8
XC3S700A-4FT256I	2,0384	134	25,86	140	6
EP3C5F256I7	2,0307	135	26,8	145	10
XA3S500E-4CPG132I	2,0200	136	24,36	128	-8
XC3S200-5FT256C	2,0196	137	25,55011	136	-1
XC3S700A-5FT256C	2,0065	138	27	147	9
EP3C16E144C8	1,9982	139	26,7	143	4
XC3S400-5PQ208C	1,9897	140	25,45	132	-8
XC3S500E-4PQ208C	1,9721	141	24,9	129	-12
XC3S200-4FT256C	1,9620	142	25,55	135	-7
XC3S200-4FT256I	1,9620	143	25,55011	137	-6
EP3C10F256C7	1,9600	144	28,1	153	9
EP2C8T144C6	1,9484	145	28,8	160	15
EP2C8F256C7	1,9410	146	28,1	154	8
XC3S400-4PQ208I	1,9374	148	25,45	134	-14
XC3S400-4PQ208C	1,9374	147	25,45	133	-14
XA3S250E-4FTG256I	1,9336	149	25,8	139	-10
EP3C10E144I7	1,9134	150	28,8	161	11
XC3S400-5FT256C	1,9081	151	27	148	-3
EP3C5U256C6	1,9047	152	29,6	166	14
EP2C8F256I8	1,8749	154	28,1	156	2
EP2C8AF256I8	1,8749	153	28,1	155	2
EP3C16M164C8	1,8742	155	28,5	157	2

Tecnología	P_Final	Orden por PF	Precio	Orden por precio	Arriba/abajo
XA3S250E-4PQG208Q	1,8690	156	26,82	146	-10
XC3S400-4FT256I	1,8592	158	27	150	-8
XC3S400-4FT256C	1,8592	157	27	149	-8
EP3C5U256I7	1,8478	159	29,6	167	8
EP3C16Q240C8	1,8452	160	29,1	163	3
XC3S400AN-4FGG400I	1,8333	161	28,92	162	1
EP3C16F256C8	1,8150	162	29,6	168	6
XC3S500E-4FT256C	1,8048	163	27,42	151	-12
XC3S400AN-5FGG400C	1,7987	164	30,18	170	6
XA3S200A-4FTG256I	1,7935	165	29,58	165	0
EP3C10U256C7	1,7912	166	30,9	172	6
XC3S500E-5PQ208C	1,7907	167	28,62	158	-9
XA3S500E-4CPG132Q	1,7741	168	28,02	152	-16
XC3S700A-4FG400C	1,7384	169	30,36	171	2
EP3C16E144C7	1,7362	170	32	177	7
XC3S500E-4PQ208I	1,7327	171	28,62	159	-12
EP3C16U256C8	1,7011	172	31,7	176	4
XA3S500E-4PQG208I	1,6994	173	29,22	164	-9
XA3S250E-4FTG256Q	1,6942	174	29,64	169	-5
XC3S500E-5FT256C	1,6304	175	31,56	173	-2
EP3C10F256C6	1,6247	176	35,2	188	12
EP3C16F484C8	1,6223	177	33,2	180	3
EP3C16M164C7	1,6221	178	34,3	184	6
EP2C8F256C6	1,6196	179	35,2	189	10
XA3S200A-4FTG256Q	1,6003	180	33,12	179	-1
XC3S500E-4FG320C	1,5914	181	31,56	174	-7
XC3S500E-4FT256I	1,5821	182	31,56	175	-7
EP3C10F256I7	1,5715	183	35,2	190	7
EP3C16F256C7	1,5686	184	35,5	191	7
XC3S700A-4FG484C	1,5611	185	33,78	182	-3
XA3S500E-4FTG256I	1,5543	186	32,16	178	-8
XA3S400A-4FTG256I	1,5404	187	34,7	185	-2
XC3S700A-4FG400I	1,5153	188	34,92	187	-1
XC3S1400A-4FT256C	1,5034	189	34,14	183	-6
XC3S700A-5FG400C	1,5033	190	36,48	195	5
EP2C15AF256C8	1,5023	191	34,7	186	-5
XA3S500E-4PQG208Q	1,4898	192	33,6	181	-11
EP3C10U256C6	1,4819	193	38,7	201	8
EP3C16U484C8	1,4814	194	36,5	196	2
EP3C16U256C7	1,4691	195	38	199	4

Tecnología	P_Final	Orden por PF	Precio	Orden por precio	Arriba/abajo
EP2C15AF256C6	1,4433	196	38,9	205	9
EP3C10U256I7	1,4332	197	38,7	202	5
XC3S500E-5FG320C	1,4330	198	36,3	193	-5
XC3S400-4FG320C	1,4282	199	35,69988	192	-7
XC3S700AN-4FGG484C	1,4082	200	37,14	198	-2
EP3C10E144A7	1,4024	201	39,5	208	7
EP3C16F484C7	1,3979	202	39,9	210	8
XC3S500E-4FG320I	1,3932	203	36,3	194	-9
XA3S400A-4FTG256Q	1,3735	204	38,88	204	0
EP3C25E144C8	1,3637	205	39,5	209	4
XA3S500E-4FTG256Q	1,3612	206	36,96	197	-9
XC3S700A-4FG484I	1,3596	207	38,82	203	-4
XC3S700A-5FG484C	1,3489	208	40,5	211	3
EP2C15AF484C8	1,3470	209	38,9	206	-3
XC3S1400A-4FT256I	1,3115	210	39,3	207	-3
XC3S1200E-4FT256C	1,3096	211	38,04	200	-11
EP2C15AF256C7	1,3091	212	41,6	215	3
XC3S1400A-5FT256C	1,3010	213	40,98	212	-1
EP3C16F256C6	1,2985	214	44,4	227	13
XC3S400-5FG320C	1,2810	215	41,05012	213	-2
EP3C16U484C7	1,2723	216	43,9	225	9
EP3C25Q240C8	1,2640	217	42,7	217	0
EP2C15AF256I8	1,2602	218	41,6	216	-2
XC3S400-4FG320I	1,2453	219	41,05012	214	-5
EP2C20F256C8	1,2348	220	42,7	218	-2
EP2C20Q240C8	1,2328	221	42,7	219	-2
XC3S700AN-4FGG484I	1,2240	222	42,72	220	-2
XC3S700AN-5FGG484C	1,2138	223	44,58	228	5
EP3C16U256C6	1,2135	224	47,5	241	17
EP3C25F256C8	1,2111	225	44,8	229	4
XC3S1000-5FT256I	1,1834	226	43,9	226	0
XC3S1200E-5FT256C	1,1821	227	43,74	222	-5
XC3S1000-4FG320C	1,1783	228	43	221	-7
EP2C15AF484C7	1,1660	229	46,7	233	4
EP3C25U256C8	1,1572	230	46,9	235	5
XC3S1200E-4FT256I	1,1451	231	43,74	223	-8
EP3C16F484C6	1,1448	232	49,9	246	14
XC3S1200E-4FG320C	1,1406	233	43,74	224	-9
EP3C10F256A7	1,1348	234	48,3	243	9
EP2C15AF484I8	1,1242	235	46,7	234	-1

Tecnología	P_Final	Orden por PF	Precio	Orden por precio	Arriba/abajo
EP2C20F484C8	1,1236	236	46,9	236	0
XC3S1000-5FT256C	1,1153	237	46,6	230	-7
XA3S400A-4FGG400I	1,1076	238	48,06	242	4
XC3S400-5FG456C	1,1039	239	47,05	237	-2
EP3C25F324C8	1,0939	240	49,3	245	5
XC3S1000-4FT256I	1,0852	242	46,6	232	-10
XC3S1000-4FT256C	1,0852	241	46,6	231	-10
XC3S1000-4FG320I	1,0746	243	47,18	240	-3
XC3S400-4FG456I	1,0725	245	47,05	239	-6
XC3S400-4FG456C	1,0725	244	47,05	238	-6
XA3S1200E-4FTG256I	1,0343	246	48,48	244	-2

Tabla 92. Clasificación de las 246 tecnologías basada en las PF

7.6.3. Sectores comerciales

A igual que el PdC global los resultados de los PdC sectoriales también se organizan en dos grupos: todos los circuitos del PdC-sector y todas las FPGAs con circuitos comunes por sector.

Entre la Tabla 94 y la Tabla 107 se encuentran las puntuaciones globales otorgadas a cada tecnología frente a cada uno de los circuitos de cada grupo y a continuación la puntuación final conseguida por sector.

Automoción-Todo el PdC

Debido al hecho de que el circuito AES128_FAST pertenece al PdC de Automoción, esta consulta sigue devolviendo un total de 9 tecnologías, aunque con una puntuación diferente a la vista en la Tabla 88 ya que solo participan 7 circuitos, Tabla 93.

bench_auto
Circuito
ADCRECV
AES128_FAST
AES_DEC
AES_ENC
SIN
SPI_BOOT
T400_CORE

Tabla 93. Circuitos seleccionados para el PdC-Automoción

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
EP3C16F484C8	AES128_FAST	8,8694	10,0000	10,0000	10,0000	10,0000	9,7629	33,2
EP3C16U484C8	AES128_FAST	7,0131	7,7830	7,5687	7,5687	8,0380	7,5866	36,5
EP3C16F484C7	AES128_FAST	10,0000	5,8824	5,4843	5,4843	6,3561	6,4595	39,9
EP2C15AF484C8	AES128_FAST	3,7842	4,1122	6,0595	6,0595	3,6119	4,6018	38,9
EP3C16U484C7	AES128_FAST	8,0262	4,0233	3,4454	3,4454	4,7108	4,4806	43,9

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
EP3C16F484C8	AES_ENC	2,5670	4,4438	2,8354	2,0514	1,8936	2,6292	33,2
EP3C16F484C8	AES_DEC	2,0192	5,1716	2,8354	2,0514	1,8956	2,5837	33,2
EP3C16U484C8	AES_ENC	2,2529	4,0040	2,4423	1,7674	1,7794	2,3341	36,5
EP3C16U484C8	AES_DEC	1,8110	4,7808	2,4423	1,7674	1,7815	2,3156	36,5
EP3C16F484C7	AES_DEC	1,9345	4,4456	2,1052	1,5239	1,6836	2,1548	39,9
EP3C16F484C7	AES_ENC	2,2204	3,6269	2,1052	1,5239	1,6816	2,1261	39,9
EP3C16F484C8	T400_CORE	2,7958	3,1529	1,7933	1,7933	1,3611	2,0763	33,2
EP3C16F484C6	AES128_FAST	7,6341	1,7935	1,0000	1,0000	2,7375	2,0643	49,9
EP2C20F484C8	AES128_FAST	1,0000	7,9451	2,1445	2,1445	1,0000	2,0538	46,9
EP2C15AF484C8	AES_ENC	1,5830	3,6911	2,1640	1,5663	1,6367	2,0052	38,9
EP2C15AF484C8	AES_DEC	1,2871	4,4963	2,1640	1,5663	1,6256	1,9986	38,9
EP3C16U484C7	AES_DEC	1,7325	4,1179	1,7756	1,2857	1,5879	1,9166	43,9
EP3C16U484C7	AES_ENC	1,9354	3,2581	1,7756	1,2857	1,5858	1,8694	43,9
EP2C15AF484C7	AES128_FAST	4,1943	1,0000	2,2260	2,2260	1,0544	1,8541	46,7
EP3C16U484C8	T400_CORE	2,4956	2,6118	1,5790	1,5790	1,3147	1,8448	36,5
EP3C16F484C7	T400_CORE	2,7024	2,1478	1,3952	1,3952	1,2750	1,7049	39,9
EP3C16F484C6	AES_DEC	1,7408	3,7251	1,3801	1,0000	1,4731	1,6750	49,9
EP2C15AF484C7	AES_DEC	1,2644	3,8872	1,5498	1,1226	1,4598	1,6568	46,7
EP2C15AF484C8	T400_CORE	1,8880	2,2610	1,4459	1,4459	1,2396	1,6173	38,9
EP2C15AF484I8	AES_DEC	1,0248	3,8872	1,5498	1,1226	1,4598	1,5886	46,7
EP3C16F484C6	AES_ENC	1,7693	2,8157	1,3801	1,0000	1,4710	1,5885	49,9
EP2C20F484C8	AES_DEC	1,0000	4,0025	1,5368	1,1132	1,4563	1,5840	46,9
EP2C15AF484C7	AES_ENC	1,2427	3,0040	1,5498	1,1226	1,4687	1,5700	46,7
EP3C16F484C8	SPI_BOOT	1,9231	1,6246	1,5631	1,5631	1,2100	1,5599	33,2
EP2C15AF484I8	AES_ENC	1,1670	3,0042	1,5498	1,1226	1,4687	1,5504	46,7
EP3C16F484C8	SIN	1,8326	1,5762	1,5631	1,5631	1,1915	1,5309	33,2
EP3C16F484C8	ADCRECV	1,8991	1,5810	1,5631	1,5631	1,0778	1,5122	33,2
EP2C20F484C8	AES_ENC	1,0000	3,1115	1,5368	1,1132	1,4652	1,5080	46,9
EP3C16U484C7	T400_CORE	2,4083	1,6940	1,2155	1,2155	1,2361	1,4943	43,9
EP2C15AF484I8	AES128_FAST	1,2042	1,0000	2,2260	2,2260	1,0544	1,4446	46,7
EP3C16U484C8	SPI_BOOT	1,7471	1,4559	1,4110	1,4110	1,1835	1,4306	36,5
EP3C16U484C8	SIN	1,6614	1,4205	1,4110	1,4110	1,1678	1,4056	36,5
EP3C16U484C8	ADCRECV	1,7599	1,4240	1,4110	1,4110	1,0647	1,3965	36,5
EP3C16F484C7	SPI_BOOT	1,8835	1,3112	1,2806	1,2806	1,1608	1,3628	39,9
EP3C16F484C7	SIN	1,7195	1,2871	1,2806	1,2806	1,1474	1,3302	39,9
EP2C15AF484C8	ADCRECV	1,6445	1,3264	1,3165	1,3165	1,0496	1,3174	38,9
EP2C15AF484C8	SPI_BOOT	1,4364	1,3504	1,3165	1,3165	1,1530	1,3113	38,9
EP2C15AF484C7	T400_CORE	1,8771	1,4103	1,1081	1,1081	1,1743	1,3072	46,7
EP3C16F484C7	ADCRECV	1,6666	1,2895	1,2806	1,2806	1,0534	1,3000	39,9
EP2C15AF484C8	SIN	1,3547	1,3236	1,3165	1,3165	1,1351	1,2868	38,9
EP3C16F484C6	T400_CORE	2,3391	1,1497	1,0000	1,0000	1,1895	1,2618	49,9
EP2C20F484C8	T400_CORE	1,5085	1,4375	1,1009	1,1009	1,1729	1,2525	46,9

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
EP3C16U484C7	SPI_BOOT	1,7097	1,1697	1,1530	1,1530	1,1386	1,2480	43,9
EP2C15AF484I8	T400_CORE	1,4850	1,4103	1,1081	1,1081	1,1743	1,2473	46,7
EP3C16U484C7	SIN	1,5573	1,1565	1,1530	1,1530	1,1275	1,2197	43,9
EP3C16U484C7	ADCRECV	1,5467	1,1579	1,1530	1,1530	1,0424	1,1994	43,9
EP2C15AF484C7	ADCRECV	1,6410	1,0789	1,0767	1,0767	1,0301	1,1615	46,7
EP2C15AF484C7	SPI_BOOT	1,4023	1,0845	1,0767	1,0767	1,1136	1,1445	46,7
EP3C16F484C6	SPI_BOOT	1,6695	1,0000	1,0000	1,0000	1,1120	1,1317	49,9
EP2C15AF484I8	ADCRECV	1,4278	1,0789	1,0767	1,0767	1,0301	1,1297	46,7
EP2C15AF484C7	SIN	1,3276	1,0783	1,0767	1,0767	1,1007	1,1281	46,7
EP2C20F484C8	ADCRECV	1,4155	1,0744	1,0716	1,0716	1,0296	1,1245	46,9
EP2C20F484C8	SPI_BOOT	1,2178	1,0810	1,0716	1,0716	1,1127	1,1096	46,9
EP3C16F484C6	SIN	1,5128	1,0000	1,0000	1,0000	1,1037	1,1080	49,9
EP2C15AF484I8	SPI_BOOT	1,1925	1,0845	1,0767	1,0767	1,1136	1,1080	46,7
EP3C16F484C6	ADCRECV	1,5905	1,0000	1,0000	1,0000	1,0292	1,1036	49,9
EP2C20F484C8	SIN	1,1607	1,0738	1,0716	1,0716	1,1023	1,0955	46,9
EP2C15AF484I8	SIN	1,1182	1,0783	1,0767	1,0767	1,1007	1,0900	46,7

Tabla 94. PG de las tecnologías con los 7 circuitos del Auto-PdC implementados (70 registros)

Tecnología	P_Final	Precio	Pines	U_logicas	Memoria	Mult	Encapsulado	Tmax	Tmin
EP3C16F484C6	1,3812	49,9	346	15408	516096	56	484FBGA	85	0
EP2C15AF484C7	1,3784	46,7	315	14448	239616	26	484FBGA	85	0
EP3C16F484C7	1,9790	39,9	346	15408	516096	56	484FBGA	85	0
EP3C16U484C7	1,7159	43,9	346	15408	516096	56	484UBGA	85	0
EP2C15AF484C8	1,8206	38,9	315	14448	239616	26	484FBGA	85	0
EP2C15AF484I8	1,2934	46,7	315	14448	239616	26	484FBGA	100	-40
EP2C20F484C8	1,3553	46,9	315	18752	239616	26	484FBGA	85	0
EP3C16F484C8	2,4279	33,2	346	15408	516096	56	484FBGA	85	0
EP3C16U484C8	2,1501	36,5	346	15408	516096	56	484UBGA	85	0

Tabla 95. PF de todas las tecnologías PdC-Automoción completo

Automoción-Todas las FPGAs

Los únicos circuitos del PdC-Automoción que se implementan en las 246 FPGAS son los denominados SIN y SPI_BOOT, por esta razón la presente búsqueda devuelve 492 registros, Tabla 96. La Tabla 97 recoge algunos registros de los 246 obtenidos para la PF de este mismo caso.

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
XC3S50A-4VQ100C	SIN	10,00	10,00	10,00	10,00	10,00	10,00	5,52
XC3S50A-4VQ100C	SPI_BOOT	10,00	10,00	10,00	10,00	10,00	10,00	5,52
XC3S50A-4VQ100I	SIN	8,67	8,66	8,66	8,66	8,80	8,69	6,36
XC3S50A-4VQ100I	SPI_BOOT	8,68	8,65	8,66	8,66	8,80	8,69	6,36
XC3S50A-5VQ100C	SIN	9,63	8,34	8,34	8,34	8,52	8,62	6,60
XC3S50A-5VQ100C	SPI_BOOT	9,44	8,28	8,34	8,34	8,51	8,57	6,60
XC3S50A-4TQ144C	SPI_BOOT	7,75	7,36	7,39	7,39	7,66	7,51	7,44
XC3S50A-4TQ144C	SIN	7,18	7,38	7,39	7,39	7,66	7,40	7,44
XC3S50A-4TQ144I	SPI_BOOT	6,72	6,35	6,39	6,39	6,76	6,52	8,58
XC3S50A-4TQ144I	SIN	6,22	6,38	6,39	6,39	6,76	6,43	8,58
XC3S50A-5TQ144C	SIN	7,28	6,12	6,13	6,13	6,53	6,42	8,94
XC3S50A-4FT256C	SPI_BOOT	6,61	6,21	6,26	6,26	6,64	6,39	8,76
XC3S50A-5TQ144C	SPI_BOOT	7,10	6,05	6,13	6,13	6,53	6,38	8,94
XC3S50A-4FT256C	SIN	6,14	6,25	6,26	6,26	6,65	6,31	8,76
XC3S50AN-4TQG144C	SPI_BOOT	6,45	6,08	6,13	6,13	5,47	6,04	8,94
XC3S50AN-4TQG144C	SIN	5,97	6,12	6,13	6,13	5,73	6,01	8,94

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
XC3S50A-4FT256I	SPI_BOOT	5,74	5,37	5,42	5,42	5,89	5,57	10,08
XC3S50A-4FT256I	SIN	5,33	5,41	5,42	5,42	5,90	5,49	10,08
XC3S50A-5FT256C	SPI_BOOT	6,25	5,09	5,17	5,17	5,66	5,45	10,56
XC3S50A-5FT256C	SIN	6,15	5,16	5,17	5,17	5,67	5,45	10,56

Tabla 96. PG de todas las FPGA ante los 2 circuitos comunes del PdC-Auto

Tecnología	P_Final	Precio	Pines	U_logicas	Memoria	Mult	Encapsulado	Tmax	Tmin
XC3S50A-4VQ100C	10,000	5,52	68	1408	55296	3	100VTQFP	85	0
XC3S50A-4VQ100I	8,6909	6,36	68	1408	55296	3	100VTQFP	100	-40
XC3S50A-5VQ100C	8,5982	6,6	68	1408	55296	3	100VTQFP	85	0
XC3S50A-4TQ144C	7,4524	7,44	108	1408	55296	3	144TQFP EP	85	0
XC3S50A-4TQ144I	6,4731	8,58	108	1408	55296	3	144TQFP EP	100	-40
XC3S50A-5TQ144C	6,3984	8,94	108	1408	55296	3	144TQFP EP	85	0
XC3S50A-4FT256C	6,3496	8,76	144	1408	55296	3	256FTBGA	85	0
XC3S50AN-4TQG144C	6,0281	8,94	108	1408	55296	3	144TQFP EP	85	0
XC3S50A-4FT256I	5,5284	10,08	144	1408	55296	3	256FTBGA	100	-40
XC3S50A-5FT256C	5,4510	10,56	144	1408	55296	3	256FTBGA	85	0
XC3S50AN-4TQG144I	5,2357	10,32	108	1408	55296	3	144TQFP EP	100	-40
XC3S50AN-5TQG144C	5,2103	10,74	108	1408	55296	3	144TQFP EP	85	0
XC3S50-4TQ144C	5,0767	9,3	97	1536	73728	50	144TQFP	85	0
XC3S50-4CP132C	5,0331	9,3	89	1536	73728	50	132CSBGA	85	0
XC3S100E-4VQ100C	4,8758	9,48	66	1920	73728	4	100VTQFP	85	0
XC3S50-5VQ100C	4,7218	10,1	63	1536	73728	50	100VTQFP	85	0
XC3S50-4VQ100C	4,5932	10,1	63	1536	73728	50	100VTQFP	85	0
XC3S50-4VQ100I	4,5932	10,1	63	1536	73728	50	100VTQFP	100	-40
XC3S50-5TQ144C	4,5463	10,7	97	1536	73728	50	144TQFP	85	0

Tabla 97. PF de todas las tecnologías ante los dos circuitos comunes del PdC-Auto

Consumo-Todo el PdC

El circuito AES128_FAST está presente entre los seleccionados para este sector comercial, Tabla 98, el total de FPGAs que se adecúan a los 10 circuitos que lo integran es de 9.

Circuito
AES128_FAST
AES_DEC
AES_ENC
SIN
SPI_BOOT
I2CSLAVETOP
RS_DEC
RS_ENCODE
UCRC_PAR
SC_CORPROC

Tabla 98. Circuitos pertenecientes al PdC-Consumo

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
EP3C16F484C8	AES128_FAST	8,8694	10,0000	10,0000	10,0000	10,0000	9,7629	33,2
EP3C16U484C8	AES128_FAST	7,0131	7,7830	7,5687	7,5687	8,0380	7,5866	36,5
EP3C16F484C7	AES128_FAST	10,0000	5,8824	5,4843	5,4843	6,3561	6,4595	39,9
EP3C16F484C8	RS_ENCODE	9,1219	6,0392	4,5408	4,5408	2,0397	4,7096	33,2
EP2C15AF484C8	AES128_FAST	3,7842	4,1122	6,0595	6,0595	3,6119	4,6018	38,9
EP3C16U484C7	AES128_FAST	8,0262	4,0233	3,4454	3,4454	4,7108	4,4806	43,9
EP3C16U484C8	RS_ENCODE	7,7677	4,9024	3,5843	3,5843	1,8064	3,8839	36,5
EP3C16F484C8	RS_DEC	3,3490	7,6333	3,9645	2,6118	2,5137	3,6696	33,2
EP3C16F484C7	RS_ENCODE	9,0526	3,9278	2,7642	2,7642	1,6064	3,3728	39,9
EP3C16U484C8	RS_DEC	2,8819	7,0142	3,6148	2,1837	2,3208	3,2637	36,5
EP2C15AF484C8	RS_ENCODE	5,0469	4,1501	2,9905	2,9905	1,3169	3,0090	38,9
EP3C16F484C7	RS_DEC	2,7000	6,4872	3,3150	1,8167	2,1554	2,9604	39,9
EP2C15AF484C8	RS_DEC	1,8881	6,4427	3,1725	1,6423	2,0417	2,6447	38,9
EP3C16F484C8	AES_ENC	2,5670	4,4438	2,8354	2,0514	1,8936	2,6292	33,2
EP3C16U484C7	RS_ENCODE	7,6942	2,9746	1,9621	1,9621	1,4108	2,6236	43,9
EP3C16U484C7	RS_DEC	2,2884	5,9677	3,0218	1,4577	1,9936	2,6049	43,9
EP3C16F484C8	AES_DEC	2,0192	5,1716	2,8354	2,0514	1,8956	2,5837	33,2
EP3C16U484C8	AES_ENC	2,2529	4,0040	2,4423	1,7674	1,7794	2,3341	36,5
EP3C16U484C8	AES_DEC	1,8110	4,7808	2,4423	1,7674	1,7815	2,3156	36,5

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
EP3C16F484C6	RS_DEC	2,3867	5,3443	2,6701	1,0272	1,7996	2,2898	49,9
EP3C16F484C7	AES_DEC	1,9345	4,4456	2,1052	1,5239	1,6836	2,1548	39,9
EP2C15AF484C7	RS_DEC	1,7220	5,4999	2,6588	1,0134	1,7641	2,1413	46,7
EP3C16F484C7	AES_ENC	2,2204	3,6269	2,1052	1,5239	1,6816	2,1261	39,9
EP3C16F484C6	AES128_FAST	7,6341	1,7935	1,0000	1,0000	2,7375	2,0643	49,9
EP2C20F484C8	AES128_FAST	1,0000	7,9451	2,1445	2,1445	1,0000	2,0538	46,9
EP2C15AF484I8	RS_DEC	1,2692	5,4979	2,6588	1,0134	1,7641	2,0144	46,7
EP2C15AF484C8	AES_ENC	1,5830	3,6911	2,1640	1,5663	1,6367	2,0052	38,9
EP2C15AF484C8	AES_DEC	1,2871	4,4963	2,1640	1,5663	1,6256	1,9986	38,9
EP2C20F484C8	RS_DEC	1,1010	6,0149	2,6479	1,0000	1,7582	1,9852	46,9
EP2C15AF484C7	RS_ENCODE	5,0464	2,3656	1,4823	1,4823	1,0066	1,9246	46,7
EP3C16U484C7	AES_DEC	1,7325	4,1179	1,7756	1,2857	1,5879	1,9166	43,9
EP3C16U484C7	AES_ENC	1,9354	3,2581	1,7756	1,2857	1,5858	1,8694	43,9
EP2C15AF484C7	AES128_FAST	4,1943	1,0000	2,2260	2,2260	1,0544	1,8541	46,7
EP2C20F484C8	RS_ENCODE	3,5122	2,4701	1,4503	1,4503	1,0000	1,7875	46,9
EP2C15AF484I8	RS_ENCODE	3,2259	2,3656	1,4823	1,4823	1,0066	1,7598	46,7
EP3C16F484C6	RS_ENCODE	7,6546	1,8312	1,0000	1,0000	1,1761	1,7515	49,9
EP3C16F484C6	AES_DEC	1,7408	3,7251	1,3801	1,0000	1,4731	1,6750	49,9
EP3C16F484C8	SC_CORPROC	1,9129	2,3208	1,5631	1,5631	1,1935	1,6689	33,2
EP2C15AF484C7	AES_DEC	1,2644	3,8872	1,5498	1,1226	1,4598	1,6568	46,7
EP3C16F484C8	UCRC_PAR	2,6699	1,5721	1,5631	1,5631	1,1933	1,6502	33,2
EP3C16F484C8	I2CSLAVETOP	2,3846	1,7468	1,5631	1,5631	1,1969	1,6487	33,2
EP2C15AF484I8	AES_DEC	1,0248	3,8872	1,5498	1,1226	1,4598	1,5886	46,7
EP3C16F484C6	AES_ENC	1,7693	2,8157	1,3801	1,0000	1,4710	1,5885	49,9
EP2C20F484C8	AES_DEC	1,0000	4,0025	1,5368	1,1132	1,4563	1,5840	46,9
EP2C15AF484C7	AES_ENC	1,2427	3,0040	1,5498	1,1226	1,4687	1,5700	46,7
EP3C16F484C8	SPI_BOOT	1,9231	1,6246	1,5631	1,5631	1,2100	1,5599	33,2
EP2C15AF484I8	AES_ENC	1,1670	3,0042	1,5498	1,1226	1,4687	1,5504	46,7
EP3C16F484C8	SIN	1,8326	1,5762	1,5631	1,5631	1,1915	1,5309	33,2
EP3C16U484C8	UCRC_PAR	2,4287	1,4176	1,4110	1,4110	1,1688	1,5161	36,5
EP3C16U484C8	SC_CORPROC	1,7267	1,9671	1,4110	1,4110	1,1650	1,5111	36,5
EP3C16U484C8	I2CSLAVETOP	2,1684	1,5450	1,4110	1,4110	1,1721	1,5087	36,5
EP2C20F484C8	AES_ENC	1,0000	3,1115	1,5368	1,1132	1,4652	1,5080	46,9
EP2C15AF484I8	AES128_FAST	1,2042	1,0000	2,2260	2,2260	1,0544	1,4446	46,7
EP3C16F484C7	UCRC_PAR	2,5078	1,2851	1,2806	1,2806	1,1479	1,4341	39,9
EP3C16U484C8	SPI_BOOT	1,7471	1,4559	1,4110	1,4110	1,1835	1,4306	36,5
EP3C16F484C7	SC_CORPROC	1,8419	1,6638	1,2806	1,2806	1,1471	1,4196	39,9
EP3C16F484C7	I2CSLAVETOP	2,1581	1,3720	1,2806	1,2806	1,1508	1,4107	39,9
EP3C16U484C8	SIN	1,6614	1,4205	1,4110	1,4110	1,1678	1,4056	36,5
EP2C15AF484C8	UCRC_PAR	1,9641	1,3215	1,3165	1,3165	1,1407	1,3869	38,9
EP2C15AF484C8	SC_CORPROC	1,4317	1,7314	1,3165	1,3165	1,1318	1,3721	38,9
EP2C15AF484C8	I2CSLAVETOP	1,6764	1,4179	1,3165	1,3165	1,1433	1,3634	38,9

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
EP3C16F484C7	SPI_BOOT	1,8835	1,3112	1,2806	1,2806	1,1608	1,3628	39,9
EP3C16F484C7	SIN	1,7195	1,2871	1,2806	1,2806	1,1474	1,3302	39,9
EP3C16U484C7	UCRC_PAR	2,2795	1,1555	1,1530	1,1530	1,1274	1,3160	43,9
EP2C15AF484C8	SPI_BOOT	1,4364	1,3504	1,3165	1,3165	1,1530	1,3113	38,9
EP3C16U484C7	I2CSLAVETOP	1,9608	1,2028	1,1530	1,1530	1,1300	1,2879	43,9
EP2C15AF484C8	SIN	1,3547	1,3236	1,3165	1,3165	1,1351	1,2868	38,9
EP3C16U484C7	SC_CORPROC	1,6607	1,3672	1,1530	1,1530	1,1232	1,2766	43,9
EP3C16U484C7	SPI_BOOT	1,7097	1,1697	1,1530	1,1530	1,1386	1,2480	43,9
EP3C16U484C7	SIN	1,5573	1,1565	1,1530	1,1530	1,1275	1,2197	43,9
EP3C16F484C6	UCRC_PAR	2,3109	1,0000	1,0000	1,0000	1,1028	1,2057	49,9
EP2C15AF484C7	I2CSLAVETOP	1,7455	1,1001	1,0767	1,0767	1,1064	1,1976	46,7
EP2C15AF484C7	UCRC_PAR	1,7725	1,0778	1,0767	1,0767	1,1043	1,1959	46,7
EP3C16F484C6	I2CSLAVETOP	2,1539	1,0000	1,0000	1,0000	1,1051	1,1894	49,9
EP2C15AF484I8	UCRC_PAR	1,6364	1,0778	1,0767	1,0767	1,1043	1,1769	46,7
EP2C20F484C8	I2CSLAVETOP	1,5186	1,0990	1,0716	1,0716	1,1056	1,1621	46,9
EP2C20F484C8	UCRC_PAR	1,4743	1,0731	1,0716	1,0716	1,1035	1,1493	46,9
EP2C15AF484C7	SC_CORPROC	1,3313	1,1764	1,0767	1,0767	1,0959	1,1475	46,7
EP2C15AF484I8	I2CSLAVETOP	1,3953	1,1001	1,0767	1,0767	1,1064	1,1451	46,7
EP2C15AF484C7	SPI_BOOT	1,4023	1,0845	1,0767	1,0767	1,1136	1,1445	46,7
EP3C16F484C6	SPI_BOOT	1,6695	1,0000	1,0000	1,0000	1,1120	1,1317	49,9
EP2C15AF484C7	SIN	1,3276	1,0783	1,0767	1,0767	1,1007	1,1281	46,7
EP2C15AF484I8	SC_CORPROC	1,1681	1,1764	1,0767	1,0767	1,0959	1,1179	46,7
EP3C16F484C6	SC_CORPROC	1,5507	1,0114	1,0000	1,0000	1,1010	1,1155	49,9
EP2C20F484C8	SC_CORPROC	1,1200	1,2145	1,0716	1,0716	1,0952	1,1133	46,9
EP2C20F484C8	SPI_BOOT	1,2178	1,0810	1,0716	1,0716	1,1127	1,1096	46,9
EP3C16F484C6	SIN	1,5128	1,0000	1,0000	1,0000	1,1037	1,1080	49,9
EP2C15AF484I8	SPI_BOOT	1,1925	1,0845	1,0767	1,0767	1,1136	1,1080	46,7
EP2C20F484C8	SIN	1,1607	1,0738	1,0716	1,0716	1,1023	1,0955	46,9
EP2C15AF484I8	SIN	1,1182	1,0783	1,0767	1,0767	1,1007	1,0900	46,7

Tabla 99. PG de las tecnologías que implementaron el PdC de consumo al completo (90 registros)

Tecnología	P_Final	Precio	Pines	U_logicas	Memoria	Mult	Encapsulado	Tmax	Tmin
EP3C16F484C6	1,4606	49,9	346	15408	516096	56	484FBGA	85	0
EP2C15AF484C7	1,4538	46,7	315	14448	239616	26	484FBGA	85	0
EP3C16F484C7	2,0831	39,9	346	15408	516096	56	484FBGA	85	0
EP3C16U484C7	1,8021	43,9	346	15408	516096	56	484UBGA	85	0
EP2C15AF484C8	1,9089	38,9	315	14448	239616	26	484FBGA	85	0
EP2C15AF484I8	1,3680	46,7	315	14448	239616	26	484FBGA	100	-40
EP2C20F484C8	1,4118	46,9	315	18752	239616	26	484FBGA	85	0
EP3C16F484C8	2,5671	33,2	346	15408	516096	56	484FBGA	85	0
EP3C16U484C8	2,2687	36,5	346	15408	516096	56	484UBGA	85	0

Tabla 100. PF de las tecnologías que implementaron el PdC de consumo al completo

Consumo-Todas las FPGAS

Este caso genera 1.240 registros para PG al conseguir implementar en todas las FPGAs 5 de los circuitos que integran este PdC, Tabla 101. El total de PG sigue siendo el total de tecnologías disponibles, 246, Tabla 102.

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
EP2C15AF256C6	I2CSLAVETOP	2,5456	1,4181	1,3165	1,3165	1,1433	1,4822	38,90
EP2C15AF256C6	UCRC_PAR	2,6347	1,3215	1,3165	1,3165	1,1407	1,4708	38,90
EP2C15AF256C6	SC_CORPROC	1,8921	1,7314	1,3165	1,3165	1,1318	1,4508	38,90
EP2C15AF256C6	SPI_BOOT	2,0713	1,3504	1,3165	1,3165	1,1530	1,4109	38,90
EP2C15AF256C6	SIN	1,8490	1,3236	1,3165	1,3165	1,1351	1,3694	38,90
EP2C5F256C6	I2CSLAVETOP	4,5908	2,7102	2,3743	2,3743	1,7613	2,6203	22,40
EP2C5F256C6	UCRC_PAR	4,3090	2,3871	2,3743	2,3743	1,7364	2,5153	22,40
EP2C5F256C6	SC_CORPROC	3,2267	3,2075	2,3743	2,3743	1,6445	2,4912	22,40
EP2C5F256C6	SPI_BOOT	3,4089	2,4804	2,3743	2,3743	1,7728	2,4287	22,40
EP2C5F256C6	SIN	3,2353	2,3906	2,3743	2,3743	1,6541	2,3530	22,40
XA3S400A-4FGG400I	UCRC_PAR	1,1047	1,0418	1,0429	1,0429	1,3863	1,1165	48,06
XA3S400A-4FGG400I	SIN	1,0987	1,0416	1,0429	1,0429	1,3922	1,1162	48,06
XA3S400A-4FGG400I	SPI_BOOT	1,0671	1,0324	1,0429	1,0429	1,4014	1,1092	48,06

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
XA3S400A-4FGG400I	I2CSLAVETOP	1,1243	1,0086	1,0429	1,0429	1,3539	1,1080	48,06
XA3S400A-4FGG400I	SC_CORPROC	1,0767	1,0017	1,0429	1,0429	1,3545	1,0970	48,06
XA3S400A-4FTG256I	SC_CORPROC	1,5381	2,0001	1,4904	1,4904	1,5229	1,5976	34,70
XA3S400A-4FTG256I	SPI_BOOT	1,5217	1,5230	1,4904	1,4904	1,5879	1,5222	34,70
XA3S400A-4FTG256I	SIN	1,5570	1,4987	1,4904	1,4904	1,5705	1,5210	34,70
XA3S400A-4FTG256I	UCRC_PAR	1,5218	1,4958	1,4904	1,4904	1,5647	1,5123	34,70
XA3S400A-4FTG256I	I2CSLAVETOP	1,4643	1,5836	1,4904	1,4904	1,5200	1,5092	34,70
XA3S400A-4FTG256Q	SC_CORPROC	1,3570	1,6140	1,3173	1,3173	1,4578	1,4083	38,88
XA3S400A-4FTG256Q	SIN	1,3831	1,3219	1,3173	1,3173	1,5015	1,3664	38,88
XA3S400A-4FTG256Q	SPI_BOOT	1,3556	1,3332	1,3173	1,3173	1,5157	1,3658	38,88
XA3S400A-4FTG256Q	UCRC_PAR	1,3584	1,3202	1,3173	1,3173	1,4957	1,3601	38,88
XA3S400A-4FTG256Q	I2CSLAVETOP	1,3061	1,3612	1,3173	1,3173	1,4558	1,3504	38,88

Tabla 101. Algunos registros PG de todas las FPGA frente al PdC-Consumo

Tecnolo.	P_Final	Precio	Pines	U_logicas	Memoria	Mult	Encap.	Tmax	Tmin
XC3S50A-4VQ100C	9,982018	5,52	68	1408	55296	3	100VTQFP	85	0
XC3S50A-4VQ100I	8,666389	6,36	68	1408	55296	3	100VTQFP	100	-40
XC3S50A-5VQ100C	8,591626	6,6	68	1408	55296	3	100VTQFP	85	0
XC3S50A-4TQ144C	7,403106	7,44	108	1408	55296	3	144TQFPEP	85	0
XC3S50A-4TQ144I	6,420832	8,58	108	1408	55296	3	144TQFPEP	100	-40
XC3S50A-5TQ144C	6,344877	8,94	108	1408	55296	3	144TQFPEP	85	0

Tecnolo.	P_Final	Precio	Pines	U_logicas	Memoria	Mult	Encap.	Tmax	Tmin
XC3S50A-4FT256C	6,293972	8,76	144	1408	55296	3	256FTBGA	85	0
XC3S50AN-4TQG144C	5,968102	8,94	108	1408	55296	3	144TQFPEP	85	0
XC3S50A-4FT256I	5,47008	10,08	144	1408	55296	3	256FTBGA	100	-40
XC3S50A-5FT256C	5,371302	10,56	144	1408	55296	3	256FTBGA	85	0
XC3S50AN-4TQG144I	5,17407	10,32	108	1408	55296	3	144TQFPEP	100	-40
XC3S50AN-5TQG144C	5,130014	10,74	108	1408	55296	3	144TQFPEP	85	0
XC3S50-4CP132C	5,083959	9,3	89	1536	73728	50	132CSBGA	85	0
XC3S50-4TQ144C	5,072247	9,3	97	1536	73728	50	144TQFP	85	0
XC3S100E-4VQ100C	4,951376	9,48	66	1920	73728	4	100VTQFP	85	0
XC3S50-5VQ100C	4,809913	10,1	63	1536	73728	50	100VTQFP	85	0
XC3S50-4VQ100C	4,663536	10,1	63	1536	73728	50	100VTQFP	85	0
XC3S50-4VQ100I	4,663536	10,1	63	1536	73728	50	100VTQFP	100	-40

Tabla 102. PF de algunas de las 246 tecnologias frente a PdC-Consumo

Comunicaciones-Todo el PdC

El PdC del sector de las comunicaciones está compuesto por 10 núcleos,Tabla 103, y no incluye el denominado AES128_FAST. A pesar de esta circunstancia el número de tecnologías capaces de implementar el PdC completo se mantiene en 9.

Circuito
SIN
SPI_BOOT
ANALYTIC_FILTER_H_A1
DDS_SYNTHESIZER
FPU
I2CSLAVETOP
RS_DEC
RS_ENCODE
UART_16750
UCRC_PAR

Tabla 103. Conjunto de circuitos de PdC-Comunicaciones

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
EP2C15AF484C7	DDS_SYNTHESIZER	1,2342	1,0829	1,0767	1,0767	1,1085	1,1142	46,7
EP2C15AF484C7	SIN	1,3276	1,0783	1,0767	1,0767	1,1007	1,1281	46,7
EP2C15AF484C7	SPI_BOOT	1,4023	1,0845	1,0767	1,0767	1,1136	1,1445	46,7
EP2C15AF484C7	UCRC_PAR	1,7725	1,0778	1,0767	1,0767	1,1043	1,1959	46,7
EP2C15AF484C7	I2CSLAVETOP	1,7455	1,1001	1,0767	1,0767	1,1064	1,1976	46,7
EP2C15AF484C7	UART_16750	1,6923	1,2663	1,1370	1,0733	1,1199	1,2397	46,7
EP2C15AF484C7	ANALYTIC_FILTER_H_A1	1,8436	1,1699	1,1408	1,4247	1,4041	1,3754	46,7
EP2C15AF484C7	RS_ENCODE	5,0464	2,3656	1,4823	1,4823	1,0066	1,9246	46,7
EP2C15AF484C7	RS_DEC	1,7220	5,4999	2,6588	1,0134	1,7641	2,1413	46,7
EP2C15AF484C7	FPU	3,1406	6,3450	1,2381	1,6219	1,9171	2,3822	46,7
EP2C15AF484C8	DDS_SYNTHESIZER	1,2724	1,3438	1,3165	1,3165	1,1482	1,2775	38,9
EP2C15AF484C8	SIN	1,3547	1,3236	1,3165	1,3165	1,1351	1,2868	38,9
EP2C15AF484C8	SPI_BOOT	1,4364	1,3504	1,3165	1,3165	1,1530	1,3113	38,9
EP2C15AF484C8	I2CSLAVETOP	1,6764	1,4179	1,3165	1,3165	1,1433	1,3634	38,9
EP2C15AF484C8	UCRC_PAR	1,9641	1,3215	1,3165	1,3165	1,1407	1,3869	38,9
EP2C15AF484C8	UART_16750	1,4681	1,8186	1,5176	1,3118	1,1606	1,4390	38,9
EP2C15AF484C8	ANALYTIC_FILTER_H_A1	2,0574	1,7087	1,5823	2,0738	1,5543	1,7812	38,9
EP2C15AF484C8	RS_DEC	1,8881	6,4427	3,1725	1,6423	2,0417	2,6447	38,9
EP2C15AF484C8	RS_ENCODE	5,0469	4,1501	2,9905	2,9905	1,3169	3,0090	38,9
EP2C15AF484C8	FPU	3,3592	7,4572	1,9826	2,3338	2,2349	3,0386	38,9
EP2C15AF484I8	DDS_SYNTHESIZER	1,0294	1,0829	1,0767	1,0767	1,1085	1,0745	46,7
EP2C15AF484I8	SIN	1,1182	1,0783	1,0767	1,0767	1,1007	1,0900	46,7
EP2C15AF484I8	SPI_BOOT	1,1925	1,0845	1,0767	1,0767	1,1136	1,1080	46,7
EP2C15AF484I8	I2CSLAVETOP	1,3953	1,1001	1,0767	1,0767	1,1064	1,1451	46,7
EP2C15AF484I8	UART_16750	1,2165	1,2665	1,1370	1,0733	1,1199	1,1606	46,7
EP2C15AF484I8	UCRC_PAR	1,6364	1,0778	1,0767	1,0767	1,1043	1,1769	46,7

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
EP2C15AF484I8	ANALYTIC_FILTER_H_A1	1,5954	1,1699	1,1408	1,4247	1,4041	1,3362	46,7
EP2C15AF484I8	RS_ENCODE	3,2259	2,3656	1,4823	1,4823	1,0066	1,7598	46,7
EP2C15AF484I8	RS_DEC	1,2692	5,4979	2,6588	1,0134	1,7641	2,0144	46,7
EP2C15AF484I8	FPU	2,8483	6,3433	1,2381	1,6219	1,8987	2,3315	46,7
EP2C20F484C8	DDS_SYNTHESIZER	1,0234	1,0794	1,0716	1,0716	1,1076	1,0704	46,9
EP2C20F484C8	SIN	1,1607	1,0738	1,0716	1,0716	1,1023	1,0955	46,9
EP2C20F484C8	SPI_BOOT	1,2178	1,0810	1,0716	1,0716	1,1127	1,1096	46,9
EP2C20F484C8	UCRC_PAR	1,4743	1,0731	1,0716	1,0716	1,1035	1,1493	46,9
EP2C20F484C8	I2CSLAVETOP	1,5186	1,0990	1,0716	1,0716	1,1056	1,1621	46,9
EP2C20F484C8	UART_16750	1,2634	1,2774	1,1289	1,0682	1,1190	1,1684	46,9
EP2C20F484C8	ANALYTIC_FILTER_H_A1	1,6891	1,1684	1,1314	1,4109	1,4009	1,3457	46,9
EP2C20F484C8	RS_ENCODE	3,5122	2,4701	1,4503	1,4503	1,0000	1,7875	46,9
EP2C20F484C8	RS_DEC	1,1010	6,0149	2,6479	1,0000	1,7582	1,9852	46,9
EP2C20F484C8	FPU	2,8172	6,8476	1,2223	1,6067	1,8915	2,3499	46,9
EP3C16F484C6	DDS_SYNTHESIZER	1,4552	1,0000	1,0000	1,0000	1,1088	1,1004	49,9
EP3C16F484C6	SIN	1,5128	1,0000	1,0000	1,0000	1,1037	1,1080	49,9
EP3C16F484C6	SPI_BOOT	1,6695	1,0000	1,0000	1,0000	1,1120	1,1317	49,9
EP3C16F484C6	UART_16750	1,7307	1,0952	1,0201	1,0000	1,1207	1,1673	49,9
EP3C16F484C6	I2CSLAVETOP	2,1539	1,0000	1,0000	1,0000	1,1051	1,1894	49,9
EP3C16F484C6	UCRC_PAR	2,3109	1,0000	1,0000	1,0000	1,1028	1,2057	49,9
EP3C16F484C6	ANALYTIC_FILTER_H_A1	2,4859	1,0000	1,0000	1,2177	1,4153	1,3377	49,9
EP3C16F484C6	RS_ENCODE	7,6546	1,8312	1,0000	1,0000	1,1761	1,7515	49,9
EP3C16F484C6	RS_DEC	2,3867	5,3443	2,6701	1,0272	1,7996	2,2898	49,9
EP3C16F484C6	FPU	4,1039	6,1235	1,0000	1,3942	2,0043	2,3404	49,9
EP3C16F484C7	DDS_SYNTHESIZER	1,5739	1,3053	1,2806	1,2806	1,1585	1,3130	39,9
EP3C16F484C7	SIN	1,7195	1,2871	1,2806	1,2806	1,1474	1,3302	39,9
EP3C16F484C7	SPI_BOOT	1,8835	1,3112	1,2806	1,2806	1,1608	1,3628	39,9
EP3C16F484C7	I2CSLAVETOP	2,1581	1,3720	1,2806	1,2806	1,1508	1,4107	39,9
EP3C16F484C7	UCRC_PAR	2,5078	1,2851	1,2806	1,2806	1,1479	1,4341	39,9
EP3C16F484C7	UART_16750	2,1596	1,7426	1,4666	1,2798	1,1718	1,5261	39,9
EP3C16F484C7	ANALYTIC_FILTER_H_A1	2,6004	1,6308	1,5166	1,9771	1,6059	1,8282	39,9
EP3C16F484C7	RS_DEC	2,7000	6,4872	3,3150	1,8167	2,1554	2,9604	39,9
EP3C16F484C7	FPU	4,3432	7,4666	1,8709	2,2269	2,4509	3,1916	39,9
EP3C16F484C7	RS_ENCODE	9,0526	3,9278	2,7642	2,7642	1,6064	3,3728	39,9
EP3C16F484C8	DDS_SYNTHESIZER	1,6713	1,6127	1,5631	1,5631	1,2086	1,5141	33,2
EP3C16F484C8	SIN	1,8326	1,5762	1,5631	1,5631	1,1915	1,5309	33,2
EP3C16F484C8	SPI_BOOT	1,9231	1,6246	1,5631	1,5631	1,2100	1,5599	33,2
EP3C16F484C8	I2CSLAVETOP	2,3846	1,7468	1,5631	1,5631	1,1969	1,6487	33,2
EP3C16F484C8	UCRC_PAR	2,6699	1,5721	1,5631	1,5631	1,1933	1,6502	33,2
EP3C16F484C8	UART_16750	2,1666	2,3942	1,9162	1,5616	1,2232	1,8017	33,2
EP3C16F484C8	ANALYTIC_FILTER_H_A1	2,9674	2,2661	2,0368	2,7419	1,7978	2,3221	33,2
EP3C16F484C8	RS_DEC	3,3490	7,6333	3,9645	2,6118	2,5137	3,6696	33,2

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
EP3C16F484C8	FPU	4,6315	8,8072	2,7480	3,0656	2,9007	3,9785	33,2
EP3C16F484C8	RS_ENCODE	9,1219	6,0392	4,5408	4,5408	2,0397	4,7096	33,2
EP3C16U484C7	DDS_SYNTHESIZER	1,4139	1,1665	1,1530	1,1530	1,1359	1,2002	43,9
EP3C16U484C7	SIN	1,5573	1,1565	1,1530	1,1530	1,1275	1,2197	43,9
EP3C16U484C7	SPI_BOOT	1,7097	1,1697	1,1530	1,1530	1,1386	1,2480	43,9
EP3C16U484C7	I2CSLAVETOP	1,9608	1,2028	1,1530	1,1530	1,1300	1,2879	43,9
EP3C16U484C7	UCRC_PAR	2,2795	1,1555	1,1530	1,1530	1,1274	1,3160	43,9
EP3C16U484C7	UART_16750	1,9593	1,4483	1,2636	1,1526	1,1485	1,3655	43,9
EP3C16U484C7	ANALYTIC_FILTER_H_A1	2,2989	1,3440	1,2817	1,6318	1,5192	1,5791	43,9
EP3C16U484C7	RS_DEC	2,2884	5,9677	3,0218	1,4577	1,9936	2,6049	43,9
EP3C16U484C7	RS_ENCODE	7,6942	2,9746	1,9621	1,9621	1,4108	2,6236	43,9
EP3C16U484C7	FPU	3,9748	6,8581	1,4749	1,8483	2,2478	2,7833	43,9
EP3C16U484C8	DDS_SYNTHESIZER	1,5036	1,4472	1,4110	1,4110	1,1816	1,3862	36,5
EP3C16U484C8	SIN	1,6614	1,4205	1,4110	1,4110	1,1678	1,4056	36,5
EP3C16U484C8	SPI_BOOT	1,7471	1,4559	1,4110	1,4110	1,1835	1,4306	36,5
EP3C16U484C8	I2CSLAVETOP	2,1684	1,5450	1,4110	1,4110	1,1721	1,5087	36,5
EP3C16U484C8	UCRC_PAR	2,4287	1,4176	1,4110	1,4110	1,1688	1,5161	36,5
EP3C16U484C8	UART_16750	1,9672	2,0432	1,6741	1,4099	1,1955	1,6253	36,5
EP3C16U484C8	ANALYTIC_FILTER_H_A1	2,6350	1,9241	1,7567	2,3301	1,6945	2,0381	36,5
EP3C16U484C8	RS_DEC	2,8819	7,0142	3,6148	2,1837	2,3208	3,2637	36,5
EP3C16U484C8	FPU	4,2399	8,0822	2,2758	2,6140	2,6585	3,5220	36,5
EP3C16U484C8	RS_ENCODE	7,7677	4,9024	3,5843	3,5843	1,8064	3,8839	36,5

Tabla 104. PG de las tecnologías que implementaron el PdC de comunicaciones completo (90 registros)

Tecnología	P_Final	Precio	Pines	U_logicas	Memoria	Mult	Encapsulado	Tmax	Tmin
EP3C16F484C6	1,4004	49,90	346	15408	516096	56	484FBGA	85	0
EP2C15AF484C7	1,4248	46,70	315	14448	239616	26	484FBGA	85	0
EP3C16F484C7	1,8330	39,90	346	15408	516096	56	484FBGA	85	0
EP3C16U484C7	1,6232	43,90	346	15408	516096	56	484UBGA	85	0
EP2C15AF484C8	1,7388	38,90	315	14448	239616	26	484FBGA	85	0
EP2C15AF484I8	1,3648	46,70	315	14448	239616	26	484FBGA	100	-40
EP2C20F484C8	1,3670	46,90	315	18752	239616	26	484FBGA	85	0
EP3C16F484C8	2,2140	33,20	346	15408	516096	56	484FBGA	85	0
EP3C16U484C8	1,9836	36,50	346	15408	516096	56	484UBGA	85	0

Tabla 105. PF de las tecnologías que implementan el PdC de comunicaciones completo

Comunicaciones-Todas las FPGAs

En esta ocasión son 6 de los 10 circuitos del PdC de comunicaciones los que se implementan con éxito en las 246 tecnologías implicadas en este estudio. La tabla *tabla_FPGA_seleccionadas* reunirá por tanto 1.476 registros, Tabla 106, y la tabla *tabla_FPGA_seleccionadas_PF* los 246 esperados, Tabla 107.

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
EP2C15AF256 C6	UART_16750	2,1531	1,8184	1,5176	1,3118	1,1606	1,5535	38,9
EP2C15AF256 C6	I2CSLAVETOP	2,5456	1,4181	1,3165	1,3165	1,1433	1,4822	38,9
EP2C15AF256 C6	UCRC_PAR	2,6347	1,3215	1,3165	1,3165	1,1407	1,4708	38,9
EP2C15AF256 C6	SPI_BOOT	2,0713	1,3504	1,3165	1,3165	1,1530	1,4109	38,9
EP2C15AF256 C6	DDS_SYNTHESIZER	1,8359	1,3438	1,3165	1,3165	1,1482	1,3747	38,9
EP2C15AF256 C6	SIN	1,8490	1,3236	1,3165	1,3165	1,1351	1,3694	38,9
EP2C5F256C6	UART_16750	4,0576	3,8084	3,1762	2,3512	1,7873	2,9032	22,4
EP2C5F256C6	I2CSLAVETOP	4,5908	2,7102	2,3743	2,3743	1,7613	2,6203	22,4
EP2C5F256C6	UCRC_PAR	4,3090	2,3871	2,3743	2,3743	1,7364	2,5153	22,4
EP2C5F256C6	SPI_BOOT	3,4089	2,4804	2,3743	2,3743	1,7728	2,4287	22,4
EP2C5F256C6	DDS_SYNTHESIZER	3,3067	2,4535	2,3743	2,3743	1,7358	2,3985	22,4
EP2C5F256C6	SIN	3,2353	2,3906	2,3743	2,3743	1,6541	2,3530	22,4
XC3S100E-4VQ100C	UART_16750	5,3961	7,3127	6,2880	5,7741	2,4594	5,1215	9,48
XC3S100E-4VQ100C	DDS_SYNTHESIZER	6,1466	5,8527	5,7729	5,7729	2,9132	5,1124	9,48
XC3S100E-4VQ100C	SPI_BOOT	5,6129	5,8795	5,7729	5,7729	2,8269	4,9949	9,48
XC3S100E-4VQ100C	I2CSLAVETOP	5,0957	6,0738	5,7729	5,7729	2,7347	4,8986	9,48
XC3S100E-4VQ100C	UCRC_PAR	5,1604	5,7904	5,7729	5,7729	2,7351	4,8644	9,48
XC3S100E-4VQ100C	SIN	5,6862	5,7987	5,7729	5,7729	2,2227	4,7595	9,48

Tecnología	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
XC3S100E-4VQ100I	DDS_SYNTHESIZER	5,3120	5,0522	4,9959	4,9959	2,6491	4,4649	10,92
XC3S100E-4VQ100I	UART_16750	4,6795	6,1522	5,3584	4,9974	2,2560	4,4470	10,92
XC3S100E-4VQ100I	SPI_BOOT	4,8696	5,0723	4,9959	4,9959	2,5750	4,3666	10,92
XC3S100E-4VQ100I	I2CSLAVETOP	4,4229	5,2089	4,9959	4,9959	2,4957	4,2793	10,92
XC3S100E-4VQ100I	UCRC_PAR	4,4802	5,0087	4,9959	4,9959	2,4961	4,2570	10,92
XC3S100E-4VQ100I	SIN	4,9283	5,0148	4,9959	4,9959	2,0521	4,1733	10,92

Tabla 106. PG de todas las FPGAs ante PdC-Comunicaciones (muestra)

Tecnología	P_Final	Precio	Pines	U_logicas	Memoria	Mult	Encapsulado	Tmax	Tmin
XC3S50A-4VQ100C	9,9786	5,52	68	1408	55296	3	100VTQFP	85	0
XC3S50A-4VQ100I	8,6622	6,36	68	1408	55296	3	100VTQFP	100	-40
XC3S50A-5VQ100C	8,6233	6,6	68	1408	55296	3	100VTQFP	85	0
XC3S50A-4TQ144C	7,3829	7,44	108	1408	55296	3	144TQFP EP	85	0
XC3S50A-4TQ144I	6,4021	8,58	108	1408	55296	3	144TQFP EP	100	-40
XC3S50A-5TQ144C	6,3584	8,94	108	1408	55296	3	144TQFP EP	85	0
XC3S50A-4FT256C	6,2902	8,76	144	1408	55296	3	256FTBGA	85	0
XC3S50AN-4TQG144C	5,9618	8,94	108	1408	55296	3	144TQFP EP	85	0
XC3S50A-4FT256I	5,4657	10,08	144	1408	55296	3	256FTBGA	100	-40
XC3S50A-5FT256C	5,3913	10,56	144	1408	55296	3	256FTBGA	85	0
XC3S50AN-4TQG144I	5,1673	10,32	108	1408	55296	3	144TQFP EP	100	-40
XC3S50AN-5TQG144C	5,1484	10,74	108	1408	55296	3	144TQFP EP	85	0
XC3S50-4TQ144C	5,1286	9,3	97	1536	73728	50	144TQFP	85	0
XC3S50-4CP132C	5,1286	9,3	89	1536	73728	50	132CSBGA	85	0
XC3S100E-4VQ100C	4,9568	9,48	66	1920	73728	4	100VTQFP	85	0
XC3S50-5VQ100C	4,8570	10,1	63	1536	73728	50	100VTQFP	85	0
XC3S50-4VQ100C	4,7193	10,1	63	1536	73728	50	100VTQFP	85	0
XC3S50-4VQ100I	4,7193	10,1	63	1536	73728	50	100VTQFP	100	-40
XC3S50-5TQ144C	4,6023	10,7	97	1536	73728	50	144TQFP	85	0

Tabla 107. PF de todas las FPGAs ante PdC-Comunicaciones (muestra)

La Tabla 108 muestra una comparativa de las PF conseguidas por las 9 tecnologías presentes en todos los análisis en cada sector.

	PF_auto	PF_con	PF_Com
EP3C16F484C8	2,4279	2,5671	2,2140
EP3C16U484C8	2,1501	2,2687	1,9836
EP3C16F484C7	1,9790	2,0831	1,8330
EP2C15AF484C8	1,8206	1,9089	1,7388
EP3C16U484C7	1,7159	1,8021	1,6232
EP3C16F484C6	1,3812	1,4606	1,4004
EP2C15AF484C7	1,3784	1,4538	1,4248
EP2C20F484C8	1,3553	1,4118	1,3670
EP2C15AF484I8	1,2934	1,3680	1,3648

Tabla 108. Confrontación de las PF de cada tecnología en los tres sectores

Aunque la puntuación de cada modelo difiere por sector, el orden prácticamente no cambia, con excepción de los modelos EP3C16F484C6 y EP2C15AF484C7 cuyos puestos se invierten ante el PdC de comunicaciones, sombreado en la Tabla 108.

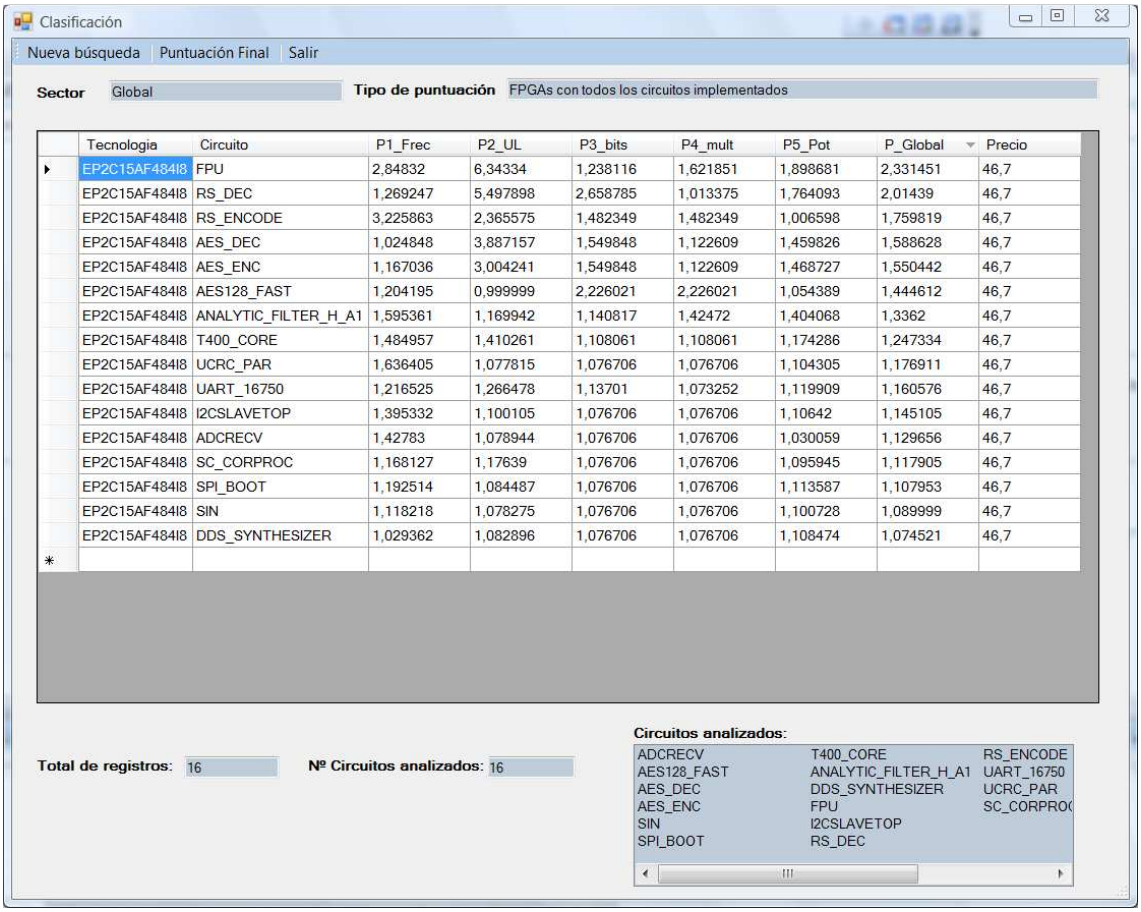
7.6.4. Selección de FPGAs

Es posible establecer infinidad de combinaciones de conjuntos de FPGA seleccionadas para diferentes criterios y mostrar de ellas sus PG y PF.

Ejemplo 1. Criterios de búsqueda: Al menos 173 pines, temperatura máxima de hasta 100ºC. Todos los circuitos del PdC.

La primera pantalla devuelve un total de 16 registros, uno por circuito e informa de cada Pk’ y de la PG del único modelo que

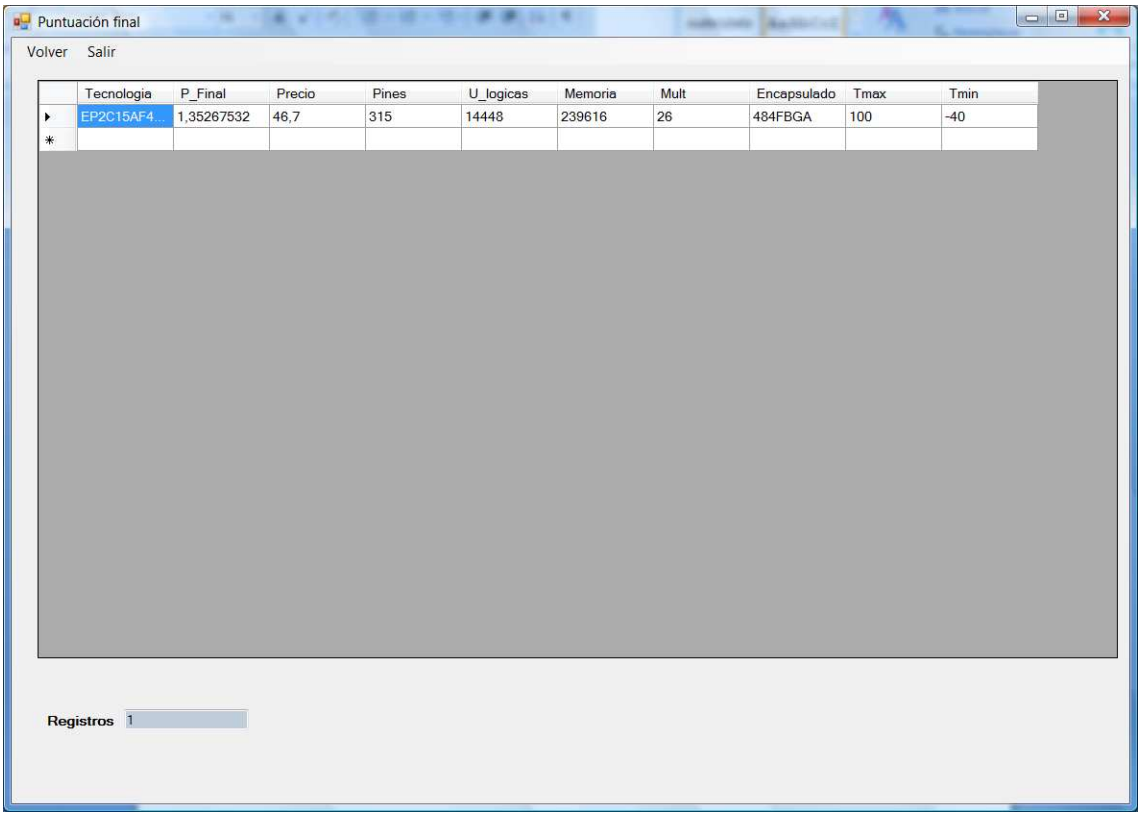
cumple todos los requisitos constructivos elegidos e implementó el PdC completo, Figura 66.



Tecnologia	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
EP2C15AF484I8	FPU	2,84832	6,34334	1,238116	1,621851	1,898681	2,331451	46,7
EP2C15AF484I8	RS_DEC	1,269247	5,497898	2,658785	1,013375	1,764093	2,01439	46,7
EP2C15AF484I8	RS_ENCODE	3,225863	2,365575	1,482349	1,482349	1,006598	1,759819	46,7
EP2C15AF484I8	AES_DEC	1,024848	3,887157	1,549848	1,122609	1,459826	1,588628	46,7
EP2C15AF484I8	AES_ENC	1,167036	3,004241	1,549848	1,122609	1,468727	1,550442	46,7
EP2C15AF484I8	AES128_FAST	1,204195	0,999999	2,226021	2,226021	1,054389	1,444612	46,7
EP2C15AF484I8	ANALYTIC_FILTER_H_A1	1,595361	1,169942	1,140817	1,42472	1,404068	1,3362	46,7
EP2C15AF484I8	T400_CORE	1,484957	1,410261	1,108061	1,108061	1,174286	1,247334	46,7
EP2C15AF484I8	UCRC_PAR	1,636405	1,077815	1,076706	1,076706	1,104305	1,176911	46,7
EP2C15AF484I8	UART_16750	1,216525	1,266478	1,13701	1,073252	1,119909	1,160576	46,7
EP2C15AF484I8	I2CSLAVETOP	1,395332	1,100105	1,076706	1,076706	1,10642	1,145105	46,7
EP2C15AF484I8	ADCRECV	1,42783	1,078944	1,076706	1,076706	1,030059	1,129656	46,7
EP2C15AF484I8	SC_CORPROC	1,168127	1,17639	1,076706	1,076706	1,095945	1,117905	46,7
EP2C15AF484I8	SPI_BOOT	1,192514	1,084487	1,076706	1,076706	1,113587	1,107953	46,7
EP2C15AF484I8	SIN	1,118218	1,078275	1,076706	1,076706	1,100728	1,089999	46,7
EP2C15AF484I8	DDS_SYNTHESIZER	1,029362	1,082896	1,076706	1,076706	1,108474	1,074521	46,7

Figura 66. Pantalla de los resultados del ejemplo 1. Orden según mayor PG.

Mediante el menú principal, *PuntuaciónFinal*, es posible avanzar hacia la segunda pantalla donde se visualiza la PF de la única tecnología obtenida como resultado del PdC global, Figura 67.



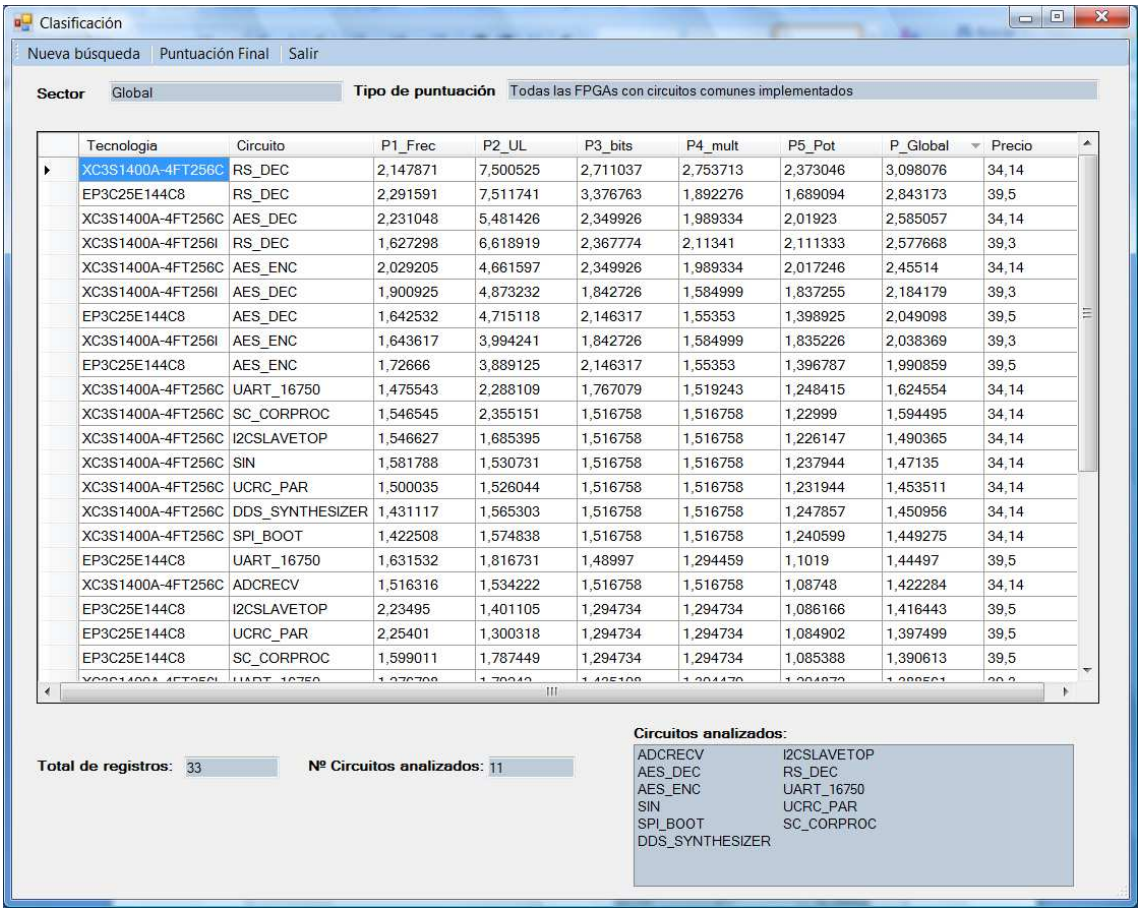
Tecnologia	P_Final	Precio	Pines	U_logicas	Memoria	Mult	Encapsulado	Tmax	Tmin
EP2C15AF4...	1,35267532	46,7	315	14448	239616	26	484FBGA	100	-40

Figura 67. PF para los resultados del ejemplo 1

Ejemplo 2. Criterios de búsqueda: Al menos 18.752 unidades lógicas y precio inferior a 39,90$, puntuar todas las FPGAs del subconjunto usando los circuitos implementados en todas.

El resultado muestra 3 FPGAs que se ajustan a todos los criterios constructivos establecidos, para todas ellas los circuitos implementados fueron 11, lo que supone 33 registros con información sobre las puntuaciones Pk' y PG de cada tecnología-circuito, Figura 68.

Este ejemplo muestra un caso en que el precio no es el principal contribuyente a la PG ya que un gran número de las FPGAs más caras ocupan las primeras posiciones en esta clasificación.



Clasificación

Nueva búsqueda | Puntuación Final | Salir

Sector Global **Tipo de puntuación** Todas las FPGAs con circuitos comunes implementados

Tecnologia	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
XC3S1400A-4FT256C	RS_DEC	2,147871	7,500525	2,711037	2,753713	2,373046	3,098076	34,14
EP3C25E144C8	RS_DEC	2,291591	7,511741	3,376763	1,892276	1,689094	2,843173	39,5
XC3S1400A-4FT256C	AES_DEC	2,231048	5,481426	2,349926	1,989334	2,01923	2,585057	34,14
XC3S1400A-4FT256I	RS_DEC	1,627298	6,618919	2,367774	2,11341	2,111333	2,577668	39,3
XC3S1400A-4FT256C	AES_ENC	2,029205	4,661597	2,349926	1,989334	2,017246	2,45514	34,14
XC3S1400A-4FT256I	AES_DEC	1,900925	4,873232	1,842726	1,584999	1,837255	2,184179	39,3
EP3C25E144C8	AES_DEC	1,642532	4,715118	2,146317	1,55353	1,398925	2,049098	39,5
XC3S1400A-4FT256I	AES_ENC	1,643617	3,994241	1,842726	1,584999	1,835226	2,038369	39,3
EP3C25E144C8	AES_ENC	1,72666	3,889125	2,146317	1,55353	1,396787	1,990859	39,5
XC3S1400A-4FT256C	UART_16750	1,475543	2,288109	1,767079	1,519243	1,248415	1,624554	34,14
XC3S1400A-4FT256C	SC_CORPROC	1,546545	2,355151	1,516758	1,516758	1,22999	1,594495	34,14
XC3S1400A-4FT256C	I2CSLAVETOP	1,546627	1,685395	1,516758	1,516758	1,226147	1,490365	34,14
XC3S1400A-4FT256C	SIN	1,581788	1,530731	1,516758	1,516758	1,237944	1,47135	34,14
XC3S1400A-4FT256C	UCRC_PAR	1,500035	1,526044	1,516758	1,516758	1,231944	1,453511	34,14
XC3S1400A-4FT256C	DDS_SYNTHESIZER	1,431117	1,565303	1,516758	1,516758	1,247857	1,450956	34,14
XC3S1400A-4FT256C	SPI_BOOT	1,422508	1,574838	1,516758	1,516758	1,240599	1,449275	34,14
EP3C25E144C8	UART_16750	1,631532	1,816731	1,48997	1,294459	1,1019	1,44497	39,5
XC3S1400A-4FT256C	ADCRECV	1,516316	1,534222	1,516758	1,516758	1,08748	1,422284	34,14
EP3C25E144C8	I2CSLAVETOP	2,23495	1,401105	1,294734	1,294734	1,086166	1,416443	39,5
EP3C25E144C8	UCRC_PAR	2,25401	1,300318	1,294734	1,294734	1,084902	1,397499	39,5
EP3C25E144C8	SC_CORPROC	1,599011	1,787449	1,294734	1,294734	1,085388	1,390613	39,5

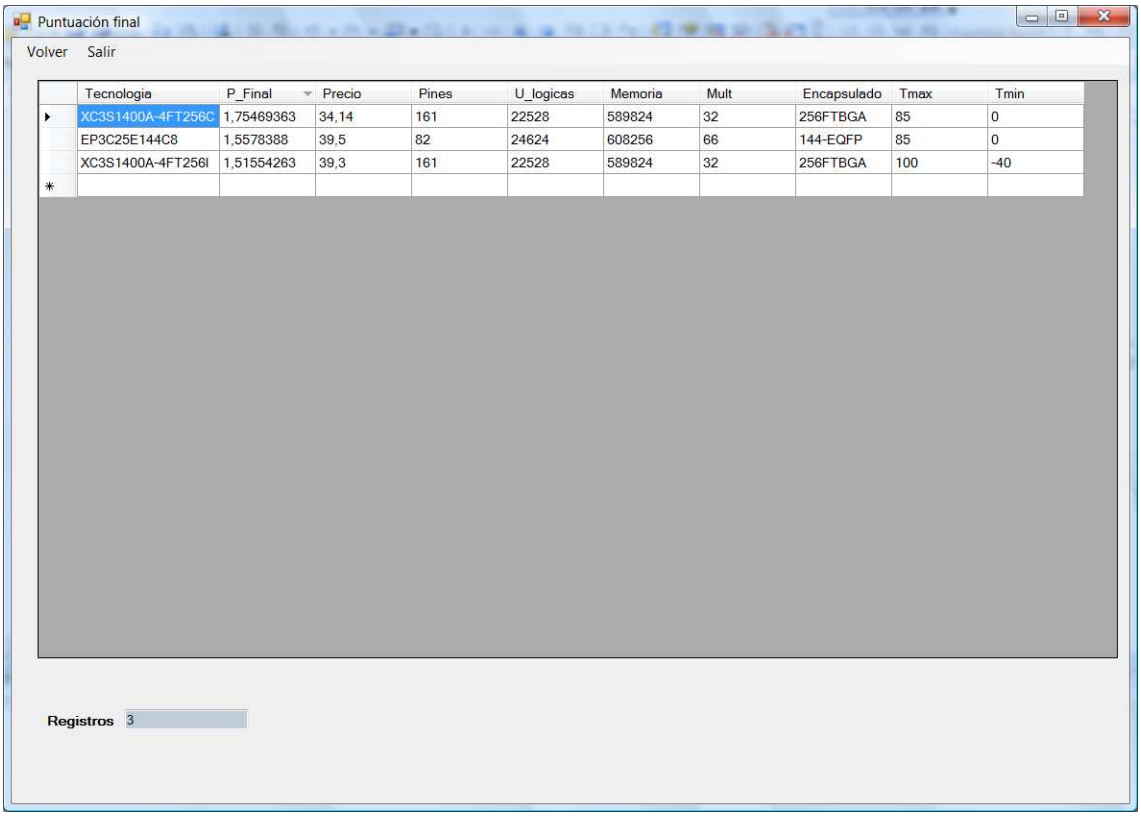
Total de registros: 33 **Nº Circuitos analizados:** 11

Circuitos analizados:

ADCRECV, AES_DEC, AES_ENC, SIN, SPI_BOOT, DDS_SYNTHESIZER, I2CSLAVETOP, RS_DEC, UART_16750, UCRC_PAR, SC_CORPROC

Figura 68. Pantalla de los resultados del ejemplo 2. Orden según PG.

En la Figura 69 se observa que las tecnologías emplazadas en los puestos segundo y tercero se venden a un precio similar, sin embargo la PF de la Cyclone3 es superior a la de la Spartan3, haciendo de la primera mejor opción.



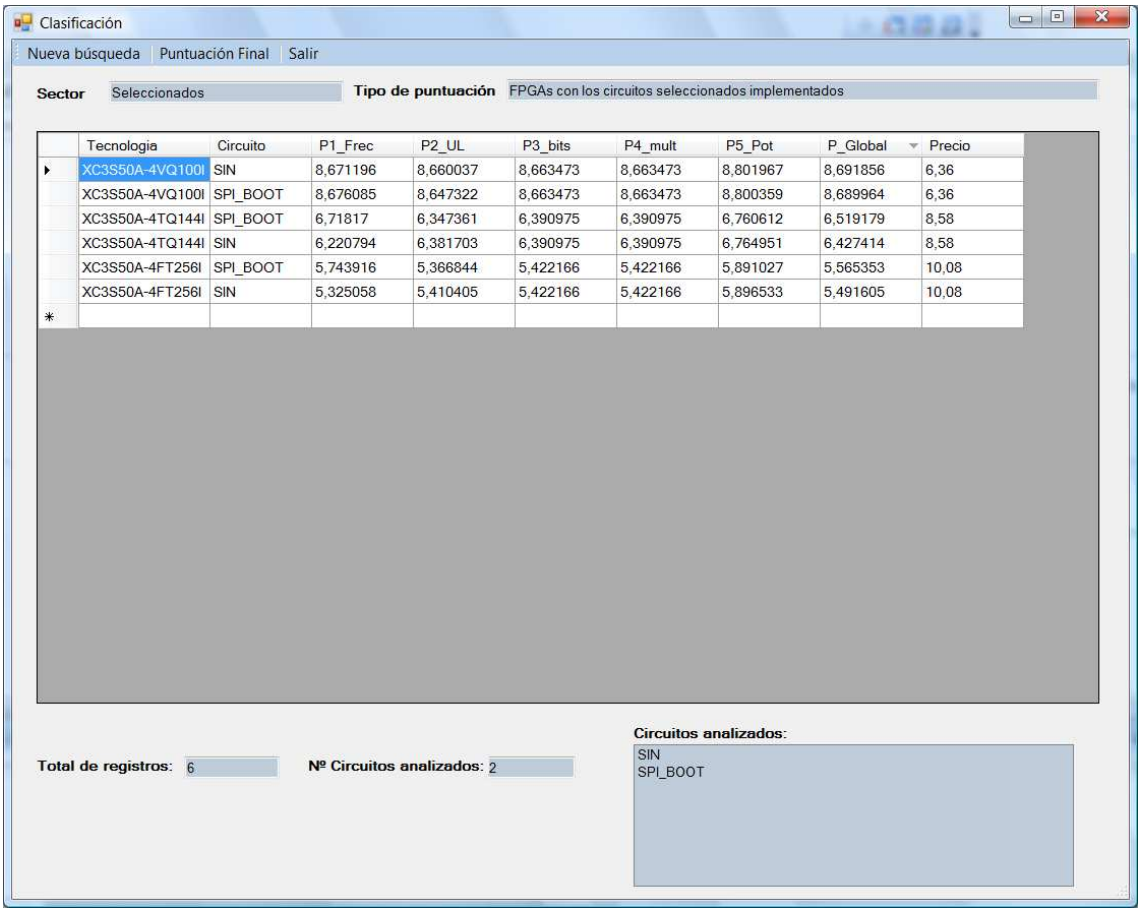
Tecnologia	P_Final	Precio	Pines	U_logicas	Memoria	Mult	Encapsulado	Tmax	Tmin
XC3S1400A-4FT256C	1,75469363	34,14	161	22528	589824	32	256FTBGA	85	0
EP3C25E144C8	1,5578388	39,5	82	24624	608256	66	144-EQFP	85	0
XC3S1400A-4FT256I	1,51554263	39,3	161	22528	589824	32	256FTBGA	100	-40

Figura 69. PF para las tecnologías resultantes del ejemplo 2

Ejemplo3. Criterios de búsqueda: Precio inferior a 10,08$ y capaz de soportar temperaturas de hasta -40ºC. Puntuación de todas las FPGA filtradas para circuitos seleccionados, SIN y SPI_BOOT.

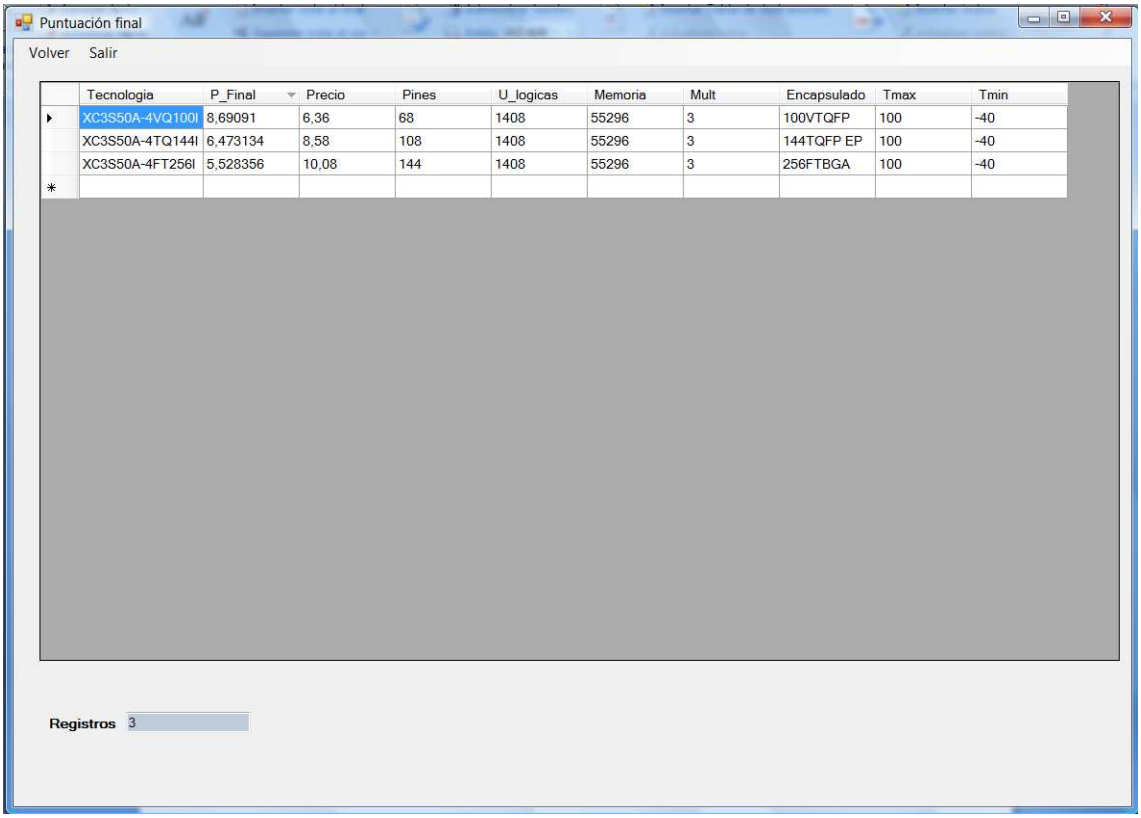
Este es otro tipo de búsqueda que se diferencia de las anteriores en que permite elegir qué circuitos deben participar en las puntuaciones. Este ejemplo se resuelve determinando incialmente el total de FPGAs que se ajustan a las características y precio, total 3 del fabricante Xilinx y a continuación se puntúan según su comportamiento ante SIN y SPI_BOOT.

La Figura 70 se corresponde con la ventana de **AsistenteFPGA** que recoge las tecnologías posibles según los criterios asi como todas sus puntuaciones normalizadas y PG. La PF estará disponible sin más que avanzar hacia la siguiente pantalla mediante el menú principal, opción *PuntuaciónFinal,* Figura 71.



Tecnologia	Circuito	P1_Frec	P2_UL	P3_bits	P4_mult	P5_Pot	P_Global	Precio
XC3S50A-4VQ100I	SIN	8,671196	8,660037	8,663473	8,663473	8,801967	8,691856	6,36
XC3S50A-4VQ100I	SPI_BOOT	8,676085	8,647322	8,663473	8,663473	8,800359	8,689964	6,36
XC3S50A-4TQ144I	SPI_BOOT	6,71817	6,347361	6,390975	6,390975	6,760612	6,519179	8,58
XC3S50A-4TQ144I	SIN	6,220794	6,381703	6,390975	6,390975	6,764951	6,427414	8,58
XC3S50A-4FT256I	SPI_BOOT	5,743916	5,366844	5,422166	5,422166	5,891027	5,565353	10,08
XC3S50A-4FT256I	SIN	5,325058	5,410405	5,422166	5,422166	5,896533	5,491605	10,08

Figura 70. Resultados del ejemplo 3 ordenados por PG.



Tecnologia	P_Final	Precio	Pines	U_logicas	Memoria	Mult	Encapsulado	Tmax	Tmin
XC3S50A-4VQ100I	8,69091	6,36	68	1408	55296	3	100VTQFP	100	-40
XC3S50A-4TQ144I	6,473134	8,58	108	1408	55296	3	144TQFP EP	100	-40
XC3S50A-4FT256I	5,528356	10,08	144	1408	55296	3	256FTBGA	100	-40

Figura 71. PF de los resultados del ejemplo 3

8. Conclusiones y trabajos futuros

8.1. Generales

El trabajo desarrollado demuestra que es posible ordenar un conjunto de FPGAs de bajo coste, atendiendo a sus prestaciones en relación con su precio unitario, y que esta clasificación es una ayuda en el momento de elegir el modelo más adecuado a un diseño.

La demostración se ha basado en la selección de un conjunto de 16 circuitos que se han implementado sobre 246 modelos de FPGAs. Los circuitos son diseños comunes entre los principales clientes de las tecnologías programables más económicas: sector de la automoción, de las comunicaciones y de la electrónica de consumo. Los núcleos elegidos tienen carácter público y están disponibles a través de internet en el principal repositorio de módulos IP, Opencores.

Una revisión del mercado de fabricantes de FPGAs determinó qué marcas tenían mayor peso, siendo Altera y Xilinx las dominantes con una cuota de mercado conjunta del 86%. Su posición privilegiada también quedó clara al analizar sus datos bursátiles, siendo las únicas empresas, dedicadas al diseño y comercialización de FPGAs, que cotizan en el Nasdaq100.

Para llevar a cabo la síntesis e implementación asi como su posterior análisis de tiempo y potencia, se ha contado con las versiones gratuitas de las herramientas ISE WebPack 10 de Xilinx y QuartusII 9.0 Web Edition de Altera, resultando ambas satisfactorias para todas las tareas y tecnologías.

La ejecución del PdC se ha efectuado configurando las herramientas según dos de las estrategias propuestas por los propios fabricantes cuyos objetivos son, por un lado conseguir la máxima frecuencia, *Tiempo*, y por otro, el equilibrio frecuencia-area-tiempo CPU la otra, *Balanced.*

La estrategia *Balanced* establecida por defecto en las herramientas gratuitas ofrecidas por Xilinx y Altera, permite conseguir buenos resultados manteniendo un equilibrio entre ocupación, frecuencia y tiempo de CPU. La mejora que supone la estrategia *tiempo* es superior al 20% en solo un 15% de los casos en ISE y de un 25% en el caso de QuartusII.

De los informes generados se ha extraído toda la información relativa a frecuencia máxima, porcentaje de ocupación de los distintos recursos lógicos y potencia de consumo estimada. Aprovechando los datos relativos al número de unidades lógicas necesarias para implementar cada circuito sobre cada tecnología, se ha tratado de establecer una equivalencia entre el *slice* de Xilinx y el *logic element* de Altera. El resultado obtenido se aleja de cualquier valor constante y se ha presentado dependiente del diseño tratado.

Relacionando los valores de frecuencia máxima, recursos sobrantes y potencia de consumo estimada con precio de venta unitario de cada tecnología, se ha otorgado un total de 5 puntuaciones a cada una por cada circuito que ha alojado con éxito. Cada una de estas calificaciones es mayor según el parámetro presentara mejor comportamiento a un mismo precio o bien a un precio menor.

La puntuación global de cada tecnología por circuito se ha calculado dando igual peso a cada una de las calificaciones individuales, lo que es consecuente con haber utilizado como objetivo de implementación la estrategia de equilibrio *Balanced.*

La confrontación de las magnitudes con el precio de mercado, ha permitido incluir en la clasificación, FPGAs con cantidades de recursos muy dispares sin que esta característica las beneficiara sobre las menos dotadas. De hecho, las puntuaciones obtenidas han favorecido en el orden, principalmente, a las tecnologías más económicas, que en esta muestra pertenecen al fabricante Xilinx.

Para conseguir la clasificación final, PF, ha sido necesario eliminar algunos de los circuitos del PdC, en concreto 9, que no se han conseguido implementar en todas las tecnologías de la muestra. Los circuitos que no han tenido un éxito general han sido aquellos demasiado exigentes con el número de terminales o de unidades lógicas. En concreto se han descartado AES128_FAST, AES_DEC, AES_ENC, RS_DEC, RS_ENCODE, FPU, T400_CORE, ADC_RECV y ANALYTIC_FILTER_H_A1, la mayoría de ellos por insuficiencia de terminales de usuario. Por lo tanto, parece interesante revisar las condiciones de selección de los IPs integrantes del PdC e incluir solo aquellos que se ajusten al modelo con menor número de entradas/salidas y de unidades lógicas.

Altera ofrece dispositivos desde 12,8$ y por debajo de ese precio Xilinx dispone de 35 modelos, a la venta desde 5,52$. Esta circunstancia origina que los productos de Xilinx ocupen las primeras 30 posiciones de la clasificación. Ya dentro de este grupo, destaca que la serie Spartan3A/AN ocupa los primeros 10 puestos y hay que tener en cuenta que la variante AN dispone además de memoria flash de configuración, un valor añadido.

Pero recordando que esta clasificación solo utiliza 7 de los circuitos del PdC y retomando por tanto los resultados finales reales, esto es, clasificación de las FPGA que han implementado los 16 circuitos, Altera gana. Solo 9 modelos han sido capaces de implementar todo el PdC por menos de 50 $, y pertenecen a este fabricante.

En cuanto a la clasificación de FPGAs según sectores comerciales, los resultados se manifiestan poco determinantes ya que la inmensa mayoría de modelos ocupan la misma posición en las tres clasificaciones. Es necesario por tanto, contar con un mayor número de núcleos y más representativos de cada sector para conseguir PdC sectoriales que ayuden a elegir la FPGA más adecuada a cada mercado.

Además de presentar un método para determinar un patrón de comparación, un sistema de puntuación y una clasificación de dispositivos FPGA, una aportación importante de esta tesis ha sido el desarrollo de las aplicaciones informáticas capaces de llevar a cabo tanto la lectura de los numerosos resultados como de efectuar el cálculo de puntuaciones, así como la clasificación y la selección ordenada de diferentes modelos de FPGAs.

La aplicación de lectura de informes, **ResultadosPdC,** abre todos los ficheros obtenidos cuya lectura hubiera sido implanteable de otro modo. Su funcionamiento ha sido adecuado para el actual PdC y se ha comprobado su flexibilidad a la hora de actualizar los circuitos integrantes del mismo.

El interfaz de la aplicación **AsistenteFPGA** se revela como de gran ayuda en el filtrado de tecnologías por características constructivas y precio. La posibilidad de ordenar las tecnologías resultantes según su PG o PF resuelve en muchos casos la pregunta “¿Qué modelo se ajusta mejor a mis necesidades y cuenta con mejor puntuación?”.

8.2. Futuros trabajos de investigación

En cuanto a la ampliación del PdC global y de los sectoriales sería interesante contar con otros módulos IP públicos. Esta idea conllevaría localizar otros repositorios a través de Internet, desarrollar módulos propios o contar con posibles donaciones procedentes de trabajos de investigación o de empresas.

Comprobado en este trabajo el método y la validez de las herramientas con una muestra de tamaño medio, parece interesante abarcar la lista completa de tecnologías FPGAs de bajo coste.

Una variante del método de cálculo de puntuaciones podría ser aquel que normalizase las puntuaciones Pk, una vez conocido el subconjunto filtrado de tecnologías, de modo que este presentara siempre calificaciones de 1 a 10.

Otra tarea interesante sería analizar el resto de fabricantes importantes, al menos Actel y Lattice, revisar sus herramientas y comprobar el funcionamiento de éstas adaptando el formato de la información temporal y de potencia al empleado en este trabajo. Posteriormente, para abordar esta tarea, sería necesario crear nuevos ficheros por lotes basados en los actuales. Tambien se deberían añadir líneas de código a las aplicaciones, dedicadas a la búsqueda de información según las palabras clave propias de estos nuevos fabricantes.

Efectuar pruebas con el PdC utilizando herramientas de terceros para cotejar los resultados de la síntesis lógica, es otro trabajo que reportaría una nueva clasificación solo de las FPGAs, independiente del entorno de desarrollo de los fabricantes.

En las aplicaciones desarrolladas sería de interés práctico:

- Efectuar las modificaciones pertinentes para permitir ponderar, por parte del usuario, cada puntuación individual, Pk’, antes de calcular PG y PF. Esta opción, permitiría, además de asignar mayor peso a un parámetro, obtener una puntuación final basada en un única Pk, por ejemplo la frecuencia versus precio.
- Averiguar si es posible establecer comunicación via web con las páginas de los distribuidores de FPGAs, de modo que la actualización de precios de los modelos evaluados estuviese automatizada.
- Desarrollar las herramientas como aplicaciones web con entornos de software abierto (lenguaje PHP, base de datos MySql o PostgreSql)
- Integrar la ejecución de los ficheros por lotes en la aplicación pudiendo modificar la muestra de FPGAs y los circuitos del PdC desde el mismo entorno

Estas labores de modificación de las aplicaciones de alto nivel, parecen un buen planteamiento de proyecto fin de carrera para los alumnos del grado en Ingeniería en Informática.

9. Referencias

1. *FPGAs and Structures ASICs: Low-Risk Soc for the Masses.* **Ed. Clarke, Altera Corporation.** 2006, Design Reuse.

2. *Study: Field Programmable logic rules.* **Bursky, Dave.** 2006.

3. **In-Stat.** *The Field-Programmable Gate Array (FPGA): Expanding Its Boundaries.* 2006.

4. **AECCouncil.** www.aecouncil.com.

5. **Dataquest, Gartner.** *Worlwide FPGA/PLD vendor revenues and rankings.* 2008-2007.

6. *ESL for FPGA.* **Milan Saini, Xilinx Inc.** : Technofocus Media, 2006.

7. *A neutral netllist of 10 combinatinal benchmark circuits and a target simulator in FORTRAN.* **Franz Brglez, Hideo Fujiwara.** Kyoto : Proceedings International sysmposium on circuits systems (ISCAS), 1985. págs. 695-698.

8. *Combinational profiles of sequential benchmark circuits.* **F.Brglez, D.Bryan, K.Kozminski.** Portland : Proceedings International Symposium on Circuits and Systems, 1989. págs. 1924-1934.

9. *Characteristics of the ITC'99 benchmark circuits in International Test Synthesis Workshop (ITSW).* **Davidson, Scott.** Santa Bárbara : s.n., 1999.

10. *ITC'99 Benchmark Circuits.* **Davidson, Scott.** Atlantic City : IEEE International Test Conference (ITC), 1999. pág. 1125.

11. *A set of benchmark for modular testing of SOCs.* **E.J. Marinissen, V. Iyengar, K. Chakrabarty.** : International Test Conference (ITC), 2002.

12. www.spec.org/results.html. 2009.

13. www.eembc.org. 2009

14. **J.J. Dongarra, P.Luszczek,A.Petitet.** The Linpack Benchmark History. 2001.

15. **Weiss, Alan R.** Dhrystone Benchmark: History, Analysis,"Scores" and recommendations. 2002.

16. *PREP Benchmarks for programmable logic devices.* **D. McCarty, D.Faria, P. Alfke.**: Custom Integrated Circuits Conference , 1993.

17. *Designing a high performance FPGA- Using the PREP Benchmark.* **W. Miller, K. Owyang.** : WESCON, 1993.

18. *PREP Benchmark reveal performance and capacity tradeoffs of programmable logic devices.* **Kliman, S.** : ASIC Conference and exhibit Proceedings, 1994.

19. www.ecs.umass.edu/ece/tessier/rcg/benchmarks.

20. **Barrientos Villar, J.M.** Síntesis de circuitos aritméticos sobre FPGAs. Una evaluación cuantitativa. : Universidad de Cádiz, 2003.

21. **Altera Co.** *White Paper: FPGA Performance Benchmarking Methodology.* 2007.

22. *White Paper: Performing Equivalent Timing Analysis Between the Altera Quartus II Software and Xilinx ISE.* 2005.

23. *Opencore Stamping and Benchmark Methodology.* 2008.

24. *How to perform meaningful benchmarks on FPGA from diferent vendors.* **Verma, S.** : Embedded systems design, 2008.

25. *Xilinx responds to Altera FPGA Benchmarks.* **Maxfield, Clive.** : DesignLine, 2008.

26. *Test Benchmark-what is the question?* **P.Reneson, M. Skoglund, E.Engström.** s.l. : IEEE International Conference on Softwre Testin Verification and Validation workshop (ICSTW'), 2008.

27. **Gartner.** *ASIC design starts to fall by 22% in '09.* : Techinsights, 2009.

28. **Opencores.** *Specification for the:WISHBONE System-on-Chip (SoC). Revision B.3.* 2002.

ANEXO A

ANEXO A.1. Ficheros de automatización de la ejecución del PdC

a) Xilinx

Fichero 1: ejecución_bench_xilinx.bat

```
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Este script llama a xilinx_balanced.bat o xilinx_time.bat según la ::
::estrategia seleccionada. Se repite por cada  uno de los circuitos que
::conforman el benchmark global. Aún así, se ubica cada resultado según sector
::aunque no se repiten los comunes.
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: El parámetro %1 debe indicar la estrategia deseada: balanced o tiempo
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Fecha: 26/06/2009
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
for %%i in (spi_boot sin adcRecv  aes_enc aes_dec aes128_fast  t400_core) do call
xilinx_%1 %%i Automocion
for %%k in (dds_synthesizer    analytic_filter_h_a1 ucrc_par  RS_dec  rs_encode fpu
i2cSlaveTop UART_16750) do call xilinx_%1 %%k Comunicaciones
for %%m in (sc_corproc) do call xilinx_%1 %%m Consumo
```

Fichero 2: Xilinx_balanced.bat

```
::Script_genérico
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
Parámetros
::     %1 Nombre del circuito a implementar
::     %2 Nombre del sector
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Se repiten todas las tareas por cada una de las 113 tecnologías seleccionadas
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Fecha: 26/06/2009
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
for %%h in (XA3S100E-4CPG132  XA3S100E-4TQG144      XA3S100E-4VQG100      XA3S1200E-
4FTG256 XA3S200A-4FTG256      XA3S250E-4CPG132      XA3S250E-4FTG256      XA3S250E-
4PQG208 XA3S250E-4TQG144      XA3S250E-4VQG100      XA3S400A-4FGG400      XA3S400A-
4FTG256 XA3S500E-4CPG132      XA3S500E-4FTG256      XA3S500E-4PQG208      XC3S1000-
4FG320  XC3S1000-4FT256       XC3S1000-5FT256       XC3S100E-4CP132       XC3S100E-
4TQ144  XC3S100E-4VQ100       XC3S100E-5CP132       XC3S100E-5TQ144       XC3S100E-
5VQ100  XC3S1200E-4FG320      XC3S1200E-4FT256      XC3S1200E-5FT256      XC3S1400A-
4FT256  XC3S1400A-5FT256      XC3S200-4FT256 XC3S200-4PQ208 XC3S200-4TQ144 XC3S200-4VQ100
        XC3S200-5FT256 XC3S200-5PQ208 XC3S200-5TQ144 XC3S200-5VQ100 XC3S200A-4FG320
        XC3S200A-4FT256       XC3S200A-4VQ100       XC3S200A-5FG320       XC3S200A-
5FT256  XC3S200A-5VQ100       XC3S200AN-4FTG256     XC3S200AN-5FTG256     XC3S250E-
4CP132  XC3S250E-4FT256       XC3S250E-4PQ208       XC3S250E-4TQ144       XC3S250E-
4VQ100  XC3S250E-5CP132       XC3S250E-5FT256       XC3S250E-5PQ208       XC3S250E-
5TQ144  XC3S250E-5VQ100       XC3S400-4FG320 XC3S400-4FG456 XC3S400-4FT256 XC3S400-4PQ208
        XC3S400-4TQ144 XC3S400-5FG320 XC3S400-5FG456 XC3S400-5FT256 XC3S400-5PQ208
        XC3S400-5TQ144 XC3S400A-4FG320       XC3S400A-4FG400       XC3S400A-4FT256
        XC3S400A-5FG320       XC3S400A-5FG400       XC3S400A-5FT256       XC3S400AN-
4FGG400 XC3S400AN-5FGG400     XC3S500E-4CP132       XC3S500E-4FG320       XC3S500E-
4FT256  XC3S500E-4PQ208       XC3S500E-4VQ100       XC3S500E-5CP132       XC3S500E-
5FG320  XC3S500E-5FT256       XC3S500E-5PQ208       XC3S50-4CP132  XC3S50-4PQ208  XC3S50-
4TQ144  XC3S50-4VQ100  XC3S50-5CP132  XC3S50-5PQ208  XC3S50-5TQ144  XC3S50-5VQ100
        XC3S50A-4FT256 XC3S50A-4TQ144 XC3S50A-4VQ100 XC3S50A-5FT256 XC3S50A-5TQ144
        XC3S50A-5VQ100 XC3S50AN-4TQG144      XC3S50AN-5TQG144      XC3S700A-4FG400
        XC3S700A-4FG484       XC3S700A-4FT256       XC3S700A-5FG400       XC3S700A-
5FG484  XC3S700A-5FT256       XC3S700AN-4FGG484     XC3S700AN-5FGG484 ) do (md
c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h
copy c:\Benchmark_Xilinx\%2\%1\  c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h
copy c:\Benchmark_estrategia_sintesis\balanced.txt
c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h

:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: FICHERO XST
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
echo set -tmpdir c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h >
c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.xst
echo set -xsthdpdir c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h >>
c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.xst
echo run -p %%h -ifn c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.prj >>
c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.xst
echo -ofn c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.ngc >>
c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.xst
echo -top %1 >> c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.xst
c:
cd c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h
copy c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.xst +
c:\%2_xilinx_balanced\%1\%%h\balanced.txt

:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: SÍNTESIS
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
xst -ifn %1.xst

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: TRANSLATE
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
ngdbuild  -nt timestamp -i -p %%h c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.ngc
c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.ngd

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: MAPPING ESTRATEGIA BALANCED
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
map     -p     %%h          -cm     area     -pr     off     -k     4     -c     100     -o
c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1_map.ncd
c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.ngd
c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.pcf

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: PLACE AND ROUTE ESTRATEGIA BALANCED
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
par   -w     -ol   std   -t   1   c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1_map.ncd
c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.ncd
c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.pcf

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: ANÁLISIS DE TIEMPO ESTÁTICO
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
trce c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.ncd  -o
c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.twr

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: ANÁLISIS DE POTENCIA
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
xpwr c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.ncd  -o
c:\RESULTADOS_XILINX\%2_xilinx_balanced\%1\%%h\%1.pwr
d:)
```

Fichero 3: Xilinx_tiempo.bat

```
::Script genérico
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Parámetros
::     %1 Nombre del circuito a implementar
::     %2 Nombre del sector
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Se repiten todas las tareas por cada una de las 113 tecnologías seleccionadas
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Fecha: 26/06/2009
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
for %%h in (XA3S100E-4CPG132  XA3S100E-4TQG144      XA3S100E-4VQG100      XA3S1200E-
4FTG256 XA3S200A-4FTG256      XA3S250E-4CPG132      XA3S250E-4FTG256      XA3S250E-
4PQG208 XA3S250E-4TQG144      XA3S250E-4VQG100      XA3S400A-4FGG400      XA3S400A-
4FTG256 XA3S500E-4CPG132      XA3S500E-4FTG256      XA3S500E-4PQG208      XC3S1000-
4FG320  XC3S1000-4FT256       XC3S1000-5FT256       XC3S100E-4CP132       XC3S100E-
4TQ144  XC3S100E-4VQ100       XC3S100E-5CP132       XC3S100E-5TQ144       XC3S100E-
5VQ100  XC3S1200E-4FG320      XC3S1200E-4FT256      XC3S1200E-5FT256      XC3S1400A-
4FT256  XC3S1400A-5FT256      XC3S200-4FT256 XC3S200-4PQ208 XC3S200-4TQ144 XC3S200-4VQ100
        XC3S200-5FT256 XC3S200-5PQ208 XC3S200-5TQ144 XC3S200-5VQ100 XC3S200A-4FG320
        XC3S200A-4FT256       XC3S200A-4VQ100       XC3S200A-5FG320       XC3S200A-
5FT256  XC3S200A-5VQ100       XC3S200AN-4FTG256     XC3S200AN-5FTG256     XC3S250E-
4CP132  XC3S250E-4FT256       XC3S250E-4PQ208       XC3S250E-4TQ144       XC3S250E-
4VQ100  XC3S250E-5CP132       XC3S250E-5FT256       XC3S250E-5PQ208       XC3S250E-
5TQ144  XC3S250E-5VQ100       XC3S400-4FG320 XC3S400-4FG456 XC3S400-4FT256 XC3S400-4PQ208
        XC3S400-4TQ144 XC3S400-5FG320 XC3S400-5FG456 XC3S400-5FT256 XC3S400-5PQ208
        XC3S400-5TQ144 XC3S400A-4FG320       XC3S400A-4FG400       XC3S400A-4FT256
        XC3S400A-5FG320       XC3S400A-5FG400       XC3S400A-5FT256       XC3S400AN-
4FGG400 XC3S400AN-5FGG400     XC3S500E-4CP132       XC3S500E-4FG320       XC3S500E-
4FT256  XC3S500E-4PQ208       XC3S500E-4VQ100       XC3S500E-5CP132       XC3S500E-
5FG320  XC3S500E-5FT256       XC3S500E-5PQ208       XC3S50-4CP132  XC3S50-4PQ208  XC3S50-
4TQ144  XC3S50-4VQ100  XC3S50-5CP132  XC3S50-5PQ208  XC3S50-5TQ144  XC3S50-5VQ100
        XC3S50A-4FT256 XC3S50A-4TQ144 XC3S50A-4VQ100 XC3S50A-5FT256 XC3S50A-5TQ144
        XC3S50A-5VQ100 XC3S50AN-4TQG144      XC3S50AN-5TQG144      XC3S700A-4FG400
        XC3S700A-4FG484       XC3S700A-4FT256       XC3S700A-5FG400       XC3S700A-
5FG484  XC3S700A-5FT256       XC3S700AN-4FGG484     XC3S700AN-5FGG484 ) do (md
c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h
copy c:\Benchmark_Xilinx\%2\%1\  c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h
copy c:\Benchmark_estrategia_sintesis\tiempo.txt
c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h

:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: FICHERO XST
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
echo set -tmpdir c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h >
c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.xst
echo set -xsthdpdir c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h >>
c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.xst
echo run -p %%h -ifn c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.prj >>
c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.xst
echo -ofn c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.ngc >>
c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.xst
echo -top %1 >> c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.xst
c:
cd c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h
copy c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.xst +
c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\tiempo.txt
```

```
:::::::::::::::::::::::::::::::::::
:: SÍNTESIS
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
xst -ifn %1.xst
:::::::::::::::::::::::::::::::::::
:: TRANSLATE
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
ngdbuild  -nt timestamp -i -p %%h c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.ngc
c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.ngd
:::::::::::::::::::::::::::::::::::
:: MAPPING ESTRATEGIA tiempo
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
map -p %%h -timing -logic_opt off -ol high -xe n -t 1 -cm area -pr off -k 4 -power off -
o c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1_map.ncd
c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.ngd
c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.pcf
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: PLACE AND ROUTE ESTRATEGIA TIEMPO
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
par -w  -pl high -rl high -xe n -t 1
c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1_map.ncd
c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.ncd
c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.pcf
:::::::::::::::::::::::::::::::::::
:: ANÁLISIS DE TIEMPO ESTÁTICO
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
trce c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.ncd  -o
c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.twr
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: ANÁLISIS DE POTENCIA
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
xpwr c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.ncd  -o
c:\RESULTADOS_XILINX\%2_XILINX_TIEMPO\%1\%%h\%1.pwr
d:)
```

b) Altera

Fichero 1: ejecucion_bench_altera.bat

```
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Este script llama a altera_balanced o altera_time según la estrategia seleccionada.
Se repite por cada uno de los circuitos que conforman el benchmark global. Aún asi, se
ubica cada resultado según sector aunque no se repiten los comunes.
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: El parámetro %1 debe indicar la estrategia deseada: balanced o tiempo
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Fecha: 26/06/2009
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
for %%i in (spi_boot sin adcRecv aes128_fast aes_enc aes_dec t400_core) do call
altera_%1 %%i Automocion
for %%k in (dds_synthesizer fpu i2cSlaveTop analytic_filter_h_a1 ucrc_par rs_encode
UART_16750 RS_dec) do call altera_%1 %%k  Comunicaciones
for %%m in (sc_corproc) do call altera_%1 %%m Consumo
```

Fichero 2: altera_balanced.bat

```
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Script Quartus II estrategia balanced
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Parámetros
::     %1 Nombre del circuito a implementar
::     %2 Nombre del sector: automocion, comunicaciones, consumo
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Se repiten todas las tareas por cada una de las 63 tecnologías seleccionadas
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Fecha: 26/06/2009
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

for %%h in (EP2C15AF256C6 EP2C15AF256C7     EP2C15AF256C8  EP2C15AF256I8  EP2C15AF484C7
        EP2C15AF484C8  EP2C15AF484I8  EP2C20F256C8   EP2C20F484C8   EP2C20Q240C8
        EP2C5F256C6    EP2C5F256C7    EP2C5F256C8    EP2C5F256I8    EP2C5Q208C7
        EP2C5Q208C8    EP2C5Q208I8    EP2C5T144C6    EP2C5T144C7    EP2C5T144C8
        EP2C5T144I8    EP2C8AF256I8   EP2C8F256C6    EP2C8F256C7    EP2C8F256C8
        EP2C8F256I8    EP2C8Q208C7    EP2C8Q208C8    EP2C8Q208I8    EP2C8T144C6
        EP2C8T144C7    EP2C8T144C8    EP2C8T144I8    EP3C10E144C7   EP3C10E144C8
        EP3C10E144I7   EP3C10F256C6   EP3C10F256C7   EP3C10F256C8   EP3C10F256I7
        EP3C10U256C6   EP3C10U256C7   EP3C10U256C8   EP3C10U256I7   EP3C16E144C7
        EP3C16E144C8   EP3C16F256C6   EP3C16F256C7   EP3C16F256C8   EP3C16F484C6
        EP3C16F484C7   EP3C16F484C8   EP3C16M164C7   EP3C16M164C8   EP3C16Q240C8
        EP3C16U256C6   EP3C16U256C7   EP3C16U256C8   EP3C16U484C7   EP3C16U484C8
        EP3C25E144C8   EP3C25E144I7   EP3C25F256C8   EP3C25F324C8   EP3C25Q240C8
        EP3C25U256C8   EP3C5E144C7    EP3C5E144C8    EP3C5E144I7    EP3C5F256C6
        EP3C5F256C7    EP3C5F256C8    EP3C5F256I7    EP3C5M164C7    EP3C5U256C6
        EP3C5U256C7    EP3C5U256C8    EP3C5U256I7    EP3C5E144A7    EP3C10E144A7
EP3C10F256A7) do (
md c:\RESULTADOS_ALTERA\%2_ALTERA_BALANCED\%1\%%h
copy c:\Benchmark_Altera\%2\%1\  c:\RESULTADOS_ALTERA\%2_ALTERA_BALANCED\%1\%%h
quartus_map c:\RESULTADOS_ALTERA\%2_ALTERA_BALANCED\%1\%%h\%1 --part=%%h --
optimize=balanced
quartus_fit c:\RESULTADOS_ALTERA\%2_ALTERA_BALANCED\%1\%%h\%1 --part=%%h --effort=auto
quartus_tan c:\RESULTADOS_ALTERA\%2_ALTERA_BALANCED\%1\%%h\%1 --tao
quartus_asm c:\RESULTADOS_ALTERA\%2_ALTERA_BALANCED\%1\%%h\%1
quartus_pow c:\RESULTADOS_ALTERA\%2_ALTERA_BALANCED\%1\%%h\%1 --no_input_file --
default_input_io_toggle_rate=1562500transitions/s --use_vectorless_estimation=off --
default_toggle_rate=1562500transitions/s)
```

Fichero 3: altera_tiempo.bat

```
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Script Quartus II estrategia tiempo. Optimizado para conseguir mayor frecuencia
(Speed y Fit Effort=standard)
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Parámetros
::     %1 Nombre del circuito a implementar
::     %2 Nombre del sector: automocion, comunicaciones, consumo
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Se repiten todas las tareas por cada una de las 63 tecnologías seleccionadas
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: Fecha: 26/06/2009
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

for %%h in (EP2C15AF256C6 EP2C15AF256C7     EP2C15AF256C8  EP2C15AF256I8  EP2C15AF484C7
       EP2C15AF484C8  EP2C15AF484I8  EP2C20F256C8   EP2C20F484C8   EP2C20Q240C8
       EP2C5F256C6    EP2C5F256C7    EP2C5F256C8    EP2C5F256I8    EP2C5Q208C7
       EP2C5Q208C8    EP2C5Q208I8    EP2C5T144C6    EP2C5T144C7    EP2C5T144C8
       EP2C5T144I8    EP2C8AF256I8   EP2C8F256C6    EP2C8F256C7    EP2C8F256C8
       EP2C8F256I8    EP2C8Q208C7    EP2C8Q208C8    EP2C8Q208I8    EP2C8T144C6
       EP2C8T144C7    EP2C8T144C8    EP2C8T144I8    EP3C10E144C7   EP3C10E144C8
       EP3C10E144I7   EP3C10F256C6   EP3C10F256C7   EP3C10F256C8   EP3C10F256I7
       EP3C10U256C6   EP3C10U256C7   EP3C10U256C8   EP3C10U256I7   EP3C16E144C7
       EP3C16E144C8   EP3C16F256C6   EP3C16F256C7   EP3C16F256C8   EP3C16F484C6
       EP3C16F484C7   EP3C16F484C8   EP3C16M164C7   EP3C16M164C8   EP3C16Q240C8
       EP3C16U256C6   EP3C16U256C7   EP3C16U256C8   EP3C16U484C7   EP3C16U484C8
       EP3C25E144C8   EP3C25E144I7   EP3C25F256C8   EP3C25F324C8   EP3C25Q240C8
       EP3C25U256C8   EP3C5E144C7    EP3C5E144C8    EP3C5E144I7    EP3C5F256C6
       EP3C5F256C7    EP3C5F256C8    EP3C5F256I7    EP3C5M164C7    EP3C5U256C6
       EP3C5U256C7    EP3C5U256C8    EP3C5U256I7    EP3C5E144A7    EP3C10E144A7
EP3C10F256A7) do (
md c:\RESULTADOS_ALTERA\%2_ALTERA_TIEMPO\%1\%%h
copy c:\Benchmark_Altera\%2\%1\  c:\RESULTADOS_ALTERA\%2_ALTERA_TIEMPO\%1\%%h
quartus_map c:\RESULTADOS_ALTERA\%2_ALTERA_TIEMPO\%1\%%h\%1 --part=%%h --optimize=speed
quartus_fit c:\RESULTADOS_ALTERA\%2_ALTERA_TIEMPO\%1\%%h\%1 --part=%%h --effort=standard
quartus_tan c:\RESULTADOS_ALTERA\%2_ALTERA_TIEMPO\%1\%%h\%1 --tao
quartus_asm c:\RESULTADOS_ALTERA\%2_ALTERA_TIEMPO\%1\%%h\%1
quartus_pow c:\RESULTADOS_ALTERA\%2_ALTERA_TIEMPO\%1\%%h\%1 --no_input_file --
default_input_io_toggle_rate=1562500transitions/s --use_vectorless_estimation=off --
default_toggle_rate=1562500transitions/s)
```

ANEXO A.2. Opciones de las estrategias

a) ISE WebPack 10: Balanced y Timing Performance

SÍNTESIS	BALANCED	TIMING PERFORMANCE
Source Option		
Automatic FSM Extraction	YES	YES
FSM Encoding Algorithm	Auto	Auto
Safe Implementation	No	No
FSM Style	lut	lut
RAM Extraction	Yes	Yes
RAM Style	Auto	Auto
ROM Extraction	Yes	Yes
Mux Style	Auto	Auto
Decoder Extraction	YES	YES
Priority Encoder Extraction	YES	YES
Shift Register Extraction	YES	YES
Logical Shifter Extraction	YES	YES
XOR Collapsing	YES	YES
ROM Style	Auto	Auto
Mux Extraction	YES	YES
Resource Sharing	YES	YES
Asynchronous To Synchronous	NO	NO
Multiplier Style	auto	auto
Automatic Register Balancing	No	Yes
Target Options		
Add IO Buffers	YES	YES
Global Maximum Fanout	500	500
Add Generic Clock Buffer(BUFG)	24	24
Register Duplication	YES	YES
Register Balancing	NO	YES
Move First FlipFlop Stage	-	YES
Move Last FlipFlop Stage	-	YES
Slice Packing	YES	YES
Optimize Instantiated Primitives	NO	NO
Use Clock Enable	Yes	Yes
Use Synchronous Set	Yes	Yes
Use Synchronous Reset	Yes	Yes
Pack IO Registers into IOBs	auto	false

SÍNTESIS	BALANCED	TIMING PERFORMANCE
Equivalent register Removal	YES	YES
General Options		
Optimization Goal	Speed	Speed
Optimization Effort	1	2
Library Search Order	COUNT2.lso	COUNT2.lso
Keep Hierarchy	NO	NO
Netlist Hierarchy	as_optimized	as_optimized
RTL Output	Yes	Yes
Global Optimization	AllClockNets	AllClockNets
Read Cores	YES	YES
Write Timing Constraints	NO	NO
Cross Clock Analysis	NO	NO
Hierarchy Separator	/	/
Bus Delimiter	<>	<>
Case Specifier	maintain	maintain
Slice Utilization Ratio	100	100
BRAM Utilization Ratio	100	100
Verilog 2001	YES	YES
Auto BRAM Packing	NO	NO
Slice Utilization Ratio Delta	5	5

MAPEADO	*BALANCED*	*TIMING PERFORMANCE*
Perform Timing-Driven Packing and Placement	**No**	**Yes**
Map Effort Level	**Medium**	**High**
Extra Effort	**None**	**Normal**
Starting Placer Cost Table (1 - 100)	1	1
Combinatorial Logic Optimization	No	No
Register Duplication	No	No
Ignore User Timing Constraints	No	No
Timing Mode	Non timing driven	Non timing driven
Trim Unconnected Signals	Yes	Yes
Replicate Logic to Allow Logic Level Reduction	Yes	Yes
Allow Logic Optimization Across Hierarchy	No	No
Map to Input Functions	4	4
Optimization Strategy (Cover Mode)	Area	Area
Generate Detailed MAP Report	No	No
Use RLOC Constraints	Yes	Yes
Pack I/O Registers/Latches into IOBs	**Off**	**For inputs and outputs**
Disable Register Ordering	No	No

MAPEADO	*BALANCED*	*TIMING PERFORMANCE*
CLB Pack Factor Percentage	100	100
Tri-state Buffer Transformation Mode (Advanced)	-	-
Map Slice Logic into Unused Block RAMs	No	No
Power Activity File	No	No
Other Map Command Line Options	-	-

EMPLAZAMIENTO E INTERCONEXIONADO	*BALANCED*	*TIMING PERFORMANCE*
Place and Route Mode	Normal place and route	Normal place and route
Place and Route Effort Level	**Standard**	**High**
Placer Effort Level (Overrides Overall Level)	**None**	**High**
Router Effort Level (Overrides Overall Level	**None**	**High**
Extra Effort (Highest PAR level only)	**None**	**Normal**
Starting Placer Cost Table (1-100)	1	1
Ignore User Timing Constraints	No	No
Timing Mode	Performance evaluation	Performance evaluation
Use Bonded I/Os	No	No
Generate Asynchronous Delay Report	No	No
Generate Clock Region Report	No	No
Generate Post-Place & Route Static Timing Report	Yes	Yes
Generate Post-Place & Route Simulation Model	No	No
Number of PAR Iterations (0 - 100)	3	3
Number of Results to Save (0 - 100)		
Save Results in Directory (.dir will be appended)		
Power Reduction		
Power Activity File		
Other Place & Route Command Line Options		

b) Quartus II 9.0 Web Edition

ANÁLISIS Y SÍNTESIS	BALANCED	SPEED
Top-level entity name	Count2	
Use Generated Physical Constraints File	Off	Off
Use smart compilation	Off	Off
Restructure Multiplexers	Auto	Auto
Create Debugging Nodes for IP Cores	Off	Off
Preserve fewer node names	On	On
Disable OpenCore Plus hardware evaluation	Off	Off
Verilog Version	Verilog_2001	Verilog_2001
VHDL Version	VHDL93	VHDL93
State Machine Processing	Auto	Auto
Safe State Machine	Off	Off
Extract Verilog State Machines	On	On
Extract VHDL State Machines	On	On
Ignore Verilog initial constructs	Off	Off
Iteration limit for constant Verilog loops	5000	5000
Iteration limit for non-constant Verilog loops	250	250
Add Pass-Through Logic to Inferred RAMs	On	On
Parallel Synthesis	Off	Off
DSP Block Balancing	Auto	Auto
NOT Gate Push-Back	On	On
Power-Up Don't Care	On	On
Remove Redundant Logic Cells	Off	Off
Remove Duplicate Registers	On	On
Ignore CARRY Buffers	Off	Off
Ignore CASCADE Buffers	Off	Off
Ignore GLOBAL Buffers	Off	Off
Ignore ROW GLOBAL Buffers	Off	Off
Ignore LCELL Buffers	Off	Off
Ignore SOFT Buffers	On	On
Limit AHDL Integers to 32 Bits	Off	Off
Optimization Technique	**Balanced**	**Speed**
Carry Chain Length	70	70
Auto Carry Chains	On	On
Auto Open-Drain Pins	On	On
Perform WYSIWYG Primitive Resynthesis	Off	Off
Auto ROM Replacement	On	On
Auto RAM Replacement	On	On
Auto Shift Register Replacement	Auto	Auto
Auto Clock Enable Replacement	On	On
Strict RAM Replacement	Off	Off
Allow Synchronous Control Signals	On	On
Force Use of Synchronous Clear Signals	Off	Off

ANÁLISIS Y SÍNTESIS	BALANCED	SPEED
Auto RAM to Logic Cell Conversion	Off	Off
Auto Resource Sharing	Off	Off
Allow Any RAM Size For Recognition	Off	Off
Allow Any ROM Size For Recognition	Off	Off
Allow Any Shift Register Size For Recognition	Off	Off
Use LogicLock Constraints during Resource Balancing	On	On
Ignore translate_off and synthesis_off directives	Off	Off
Timing-Driven Synthesis	Off	Off
Show Parameter Settings Tables in Synthesis Report	On	On
Ignore Maximum Fan-Out Assignments	Off	Off
Synchronization Register Chain Length	2	2
PowerPlay Power Optimization	Normal compilation	Normal compilation
HDL message level	Level2	Level2
Suppress Register Optimization Related Messages	Off	Off
Number of Removed Registers Reported in Synthesis Report	100	100
Number of Inverted Registers Reported in Synthesis Report	100	100
Clock MUX Protection	On	On
Auto Gated Clock Conversion	Off	Off
Block Design Naming	Auto	Auto
SDC constraint protection	Off	Off
Synthesis Effort	Auto	Auto
Allows Asynchronous Clear Usage For Shift Register Replacement	On	On
Analysis & Synthesis Message Level	Medium	Medium

MAPEADO, EMPLAZAMIENTO Y RUTEADO	BALANCED	SPEED
Minimum Core Junction Temperature	0	0
Maximum Core Junction Temperature	85	85
Fit Attempts to Skip	0	0
Use smart compilation	Off	Off
Use TimeQuest Timing Analyzer	Off	Off
Router Timing Optimization Level	Normal	Normal
Placement Effort Multiplier	1.0	1.0
Router Effort Multiplier	1.0	1.0
Always Enable Input Buffers	Off	Off
Optimize Hold Timing	IO Paths and Minimum TPD Paths	IO Paths and Minimum TPD

MAPEADO, EMPLAZAMIENTO Y RUTEADO	BALANCED	SPEED
		Paths
Optimize Multi-Corner Timing	Off	Off
PowerPlay Power Optimization	Normal compilation	Normal compilation
Optimize Timing	Normal compilation	Normal compilation
Optimize Timing for ECOs	Off	Off
Regenerate full fit report during ECO compiles	Off	Off
Optimize IOC Register Placement for Timing	On	On
Limit to One Fitting Attempt	Off	Off
Final Placement Optimizations	Automatically	Automatically
Fitter Aggressive Routability Optimizations	Automatically	Automatically
Fitter Initial Placement Seed	1	1
PCI I/O	Off	Off
Weak Pull-Up Resistor	Off	Off
Enable Bus-Hold Circuitry	Off	Off
Auto Global Memory Control Signals	Off	Off
Auto Packed Registers	Auto	Auto
Auto Delay Chains	On	On
Auto Merge PLLs	On	On
Ignore PLL Mode When Merging PLLs	Off	Off
Perform Physical Synthesis for Combinational Logic for Fitting	Off	Off
Perform Physical Synthesis for Combinational Logic for Performance	Off	Off
Perform Register Duplication for Performance	Off	Off
Perform Logic to Memory Mapping for Fitting	Off	Off
Perform Register Retiming for Performance	Off	Off
Perform Asynchronous Signal Pipelining	Off	Off
Fitter Effort	Auto Fit	Standard Fit
Physical Synthesis Effort Level	Normal	Normal
Auto Global Clock	On	On
Auto Global Register Control Signals	On	On
Stop After Congestion Map Generation	Off	Off
Save Intermediate Fitting Results	Off	Off
Force Fitter to Avoid Periphery Placement Warnings	Off	Off

Anexo B: Aplicaciones

Anexo B.1.ResultadosPdC

Form1.vb

```
Imports System.Data.OleDb
Public Class Form1
    Public Structure Registro
        Public Fabricante As String
        Public Circuito As String
        Public Estrategias As String
        Public Tecnologia As String
        Public Sector As String
        Public UL_usados As Integer
        Public UL_existentes As Integer
        Public Pines_usados As Integer
        Public Pines_existentes As Integer
        Public Bits_usados As Integer
        Public Bits_existentes As Integer
        Public Mult_9_usados As Integer
        Public Mult_9_existentes As Integer
        Public potencia As Single
        Public P2P As Single
        Public Tsu As Single
        Public Th As Single
        Public Tco As Single
        Public T_AandS As String
        Public T_Fit As String
        Public Pico_Memo As String
    End Structure

    Public Structure Fallo
        Public fallo_Circuito As String
        Public fallo_Estrategias As String
        Public fallo_Tecnologia As String
        Public fallo_implementacion As String
        Public fallo_potencia As String
        Public fallo_tiempo As String
        Public fallo_flow As String
    End Structure

    Public Structure Sobredimension
        Public sobre_fabricante As String
        Public sobre_circuito As String
        Public sobre_estrategia As String
        Public sobre_tecnologia As String
        Public sobre_sector As String
        Public sobre_ULs_utilizados As Single
        Public sobre_ULs_disponibles As Single
        Public sobre_IOs_utilizados As Single
        Public sobre_IOs_disponibles As Single
    End Structure


    Private Sub ToolStripButton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ToolStripButton1.Click


        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ' TABLAS DATOSREPORTS, FALLOS, SOBREDIMENSIONADOS'
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        Dim i, j, k As Integer
        Dim CompruebaFicheroExiste As Boolean = False
        Dim sector_altera As String
        Dim estrategia_altera As String
        Dim fabricante_altera As String
        Dim RegistroAltera As New Registro
```

```
        Dim RegistroFalloAltera As New Fallo
        Dim RegistroSobreAltera As New Sobredimension


 'FolderBrowserDialog1.showdialog() devuelve un valor numerico segun el boton que se pulse
 '(ACEPTAR=1/CANCELAR=2)
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
´´´´´´´´´´
        Dim ruta_valida As Integer
        ruta_valida = FolderBrowserDialog1.ShowDialog()

 'Lee la ruta y comprueba que se seleccíonó una y se pulsó ACEPTAR
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
´´´´´´´´´´
        Dim ruta_path As String
        ruta_path = FolderBrowserDialog1.SelectedPath

        If ruta_path <> "" And ruta_valida <> 2 Then
            TextBox_ruta.Text = ruta_path
        End If

'Comprueba si en la ruta seleccionada estan las carpetas RESULTADOS_ALTERA ó RESULTADOS_XILINX,
'creadas por los scripts,si es asi se da por válida la ruta
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
´´´´´´´´´´
        Dim ruta_Altera As String
        Dim ruta_Xilinx As String
        ruta_Altera = ruta_path + "\RESULTADOS_ALTERA"
        ruta_Xilinx = ruta_path + "\RESULTADOS_XILINX"
        If ((My.Computer.FileSystem.DirectoryExists(ruta_Altera)) And
(My.Computer.FileSystem.DirectoryExists(ruta_Xilinx))) Then

'Si se opta por realizar nueva lectura y nuevos calculos, se borra el contenido de las tablas.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            DatosReportsTableAdapter.DeleteQuery()
            FallosTableAdapter.DeleteQuery()
            SobredimensionadosTableAdapter.DeleteQuery()
            PuntuacionesTableAdapter.DeleteQuery()
            Fallos_puntuacionTableAdapter.DeleteQuery()
            Puntuaciones_min_maxTableAdapter.DeleteQuery()
            NormalizadosTableAdapter.DeleteQuery()
            CircuitoTableAdapter.DeleteQuery()
            Circuitos_implementadosTableAdapter.DeleteQuery()
            Bench_autoTableAdapter.DeleteQuery()
            Bench_comunicacionesTableAdapter.DeleteQuery()
            Bench_consumoTableAdapter.DeleteQuery()

                         Me.DatosReportsTableAdapter.Fill(Resultados_benchamrkDataSet.DatosReports)
             Me.SobredimensionadosTableAdapter.Fill(Resultados_benchamrkDataSet.Sobredimensionados)
            Me.FallosTableAdapter.Fill(Resultados_benchamrkDataSet.Fallos)

            TextBox2.Text = DatosReportsBindingSource.Count
            TextBox3.Text = FallosBindingSource.Count
            TextBox4.Text = SobredimensionadosBindingSource.Count


            '*************
            '* VERDADERO *
            '*************

            'Obtiene el nombre de las carpetas del siguiente nivel(Sector_Fabricante_Estrategia)
            'Y el numero total de las mismas.En principio deben ser 6
            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
            Dim DirectoriosAltera As System.Collections.ObjectModel.ReadOnlyCollection(Of String)
            Dim NumDirAltera As Integer
            DirectoriosAltera = My.Computer.FileSystem.GetDirectories(ruta_Altera)
            NumDirAltera = DirectoriosAltera.Count

            'Obtiene el nombre de todas las carpetas dentro de (Sector_Fabricante_Estrategia)
            'RESULTADOS_ALTERA\Sector_Fabricante_Estrategia\Circuito1
            'Es el nombre de los circuitos analizados
            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
            Dim DirectoriosAlteraCircuitos As System.Collections.ObjectModel.ReadOnlyCollection(Of
String)
            Dim NumDirAlteraCircuitos As Integer
            For i = 0 To NumDirAltera - 1
                DirectoriosAlteraCircuitos =
My.Computer.FileSystem.GetDirectories(DirectoriosAltera.Item(i))
                NumDirAlteraCircuitos = DirectoriosAlteraCircuitos.Count

                'Obtiene el nombre de todas las carpetas dentro de cada circuito
                'RESULTADOS_ALTERA\Sector_Fabricante_Estrategia\Circuito1\FPGA1
                'Es el nombre de las tecnologias objetivo
                '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´'''''''''''''''''''''''''
                Dim DirectoriosAlteraTecnologias As
System.Collections.ObjectModel.ReadOnlyCollection(Of String)
```

```
                Dim NumDirAlteraTecnologias As Integer
                For j = 0 To NumDirAlteraCircuitos - 1
                    DirectoriosAlteraTecnologias = My.Computer.FileSystem.GetDirectories_
                    (DirectoriosAlteraCircuitos.Item(j))
                    NumDirAlteraTecnologias = DirectoriosAlteraTecnologias.Count

              'Ya está colocado dentro de la ruta donde se encuentran los reports a leer
             'Por cada report a leer hay que asociarle la estrategia, fabricante, sector
                    '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                    For k = 0 To NumDirAlteraTecnologias - 1
                        Dim circuito_altera As String
                        Dim tecnologia_altera As String
                        Dim ver_datos() As String
                        ver_datos = datos(DirectoriosAlteraTecnologias(k))

                        sector_altera = ver_datos(0)
                        fabricante_altera = ver_datos(1)
                        estrategia_altera = ver_datos(2)
                        circuito_altera = ver_datos(3)
                        tecnologia_altera = ver_datos(4)

                        RegistroAltera.Fabricante = fabricante_altera.ToUpper
                        RegistroAltera.Circuito = circuito_altera.ToUpper
                        RegistroAltera.Estrategias = estrategia_altera.ToUpper
                        RegistroAltera.Tecnologia = tecnologia_altera
                        RegistroAltera.Sector = sector_altera.ToUpper

                        RegistroFalloAltera.fallo_Circuito = circuito_altera.ToUpper
                           RegistroFalloAltera.fallo_Estrategias = _
                         estrategia_altera.ToUpper
                        RegistroFalloAltera.fallo_Tecnologia = tecnologia_altera

                        RegistroSobreAltera.sobre_estrategia = estrategia_altera.ToUpper
                        RegistroSobreAltera.sobre_circuito = circuito_altera.ToUpper
                        RegistroSobreAltera.sobre_tecnologia = tecnologia_altera

            'Abrir ficheros de resultados
            '----------------------------------------------------------------------------
    'El nombre de los ficheros coincide con el del circuito que se está analizando y que
    'está almacenado en circuito_altera. En cada caso se le añade la extensión adecuada.


                        '''''''''''''''''
                        ' 1. FLOW REPORT'
                        '''''''''''''''''
                     ' Incluye toda la informacion relativo a los recursos empleados asi
                     'como el tiempo de CPU y memoria utilizados en cada proceso.

                        'Determinación del nombre y de la ruta del FLOW.RPT
                        '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                        Dim NombreFicheroFlowAltera As String
                        Dim PathFicheroFlowAltera As String

               NombreFicheroFlowAltera = circuito_altera + ".flow.rpt"
                 PathFicheroFlowAltera = DirectoriosAlteraTecnologias.Item(k) + "\" +_
               NombreFicheroFlowAltera

               'Comprobacion de la existencia del fichero FLOW
               '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
               CompruebaFicheroExiste = My.Computer.FileSystem.FileExists(PathFicheroFlowAltera)

                        If CompruebaFicheroExiste = True Then

                            '*************
                            '* VERDADERO *
                            '*************
                            'Busqueda de recursos logicos utilizados y disponibles
                            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                'Se llama a la funcion de busqueda de informacion "busqueda_valor_flow"
                            '´´´´´´
                            ' AREA´
                            '´´´´´´
                            Dim LEs(1) As Single
                            Dim Pines(1) As Single
                            Dim bits(1) As Single
                            Dim mult() As Single

     LEs = busqueda_valor(PathFicheroFlowAltera, "TOTAL LOGIC ELEMENTS", ";", "/", "(")
     Pines = busqueda_valor(PathFicheroFlowAltera, "TOTAL PINS", ";", "/", "(")

'Si el circuito no se ajusta a los recursos logicos disponibles se incluirá aparte
                        'TABLA: Sobredimensionados de la base de datos.
'..........................................................................
                            If (LEs(0) < LEs(1) And Pines(0) < Pines(1)) Then

                                RegistroAltera.UL_usados = LEs(0)
```

```
                                RegistroAltera.UL_existentes = LEs(1)
                                RegistroAltera.Pines_usados = Pines(0)
                                RegistroAltera.Pines_existentes = Pines(1)

'Sigue leyendo el fichero si no ha habido sobreutilizacion de los recursos lógicos
'A continuación comprueba que no se rebasa el limite de pines disponibles
'..........................................................................
                        'Lee los bits utilizados de los bloques de memoria
                        '..............................................................
       bits = busqueda_valor(PathFicheroFlowAltera, "TOTAL MEMORY BITS", ";", "/", "(")
                      RegistroAltera.Bits_usados = bits(0)
                                RegistroAltera.Bits_existentes = bits(1)

               'Lee los multiplicadores de 9 bits utilizados de los bloques de memoria
               '..............................................................
                                mult = busqueda_valor(PathFicheroFlowAltera,_
                                "EMBEDDED MULTIPLIER 9-BIT ELEMENTS", ";", "/", "(")
                                If (mult(0) = 0 And mult(1) = 0) Then
                                    mult = busqueda_valor(PathFicheroFlowAltera,_
                                   "EMBEDDED MULTIPLIER 18-BIT ELEMENTS", ";", "/", "(")
                                End If
                                RegistroAltera.Mult_9_usados = mult(0)
                                RegistroAltera.Mult_9_existentes = mult(1)
                            Else
                                RegistroSobreAltera.sobre_ULs_disponibles = LEs(1)
                                RegistroSobreAltera.sobre_ULs_utilizados = LEs(0)
                                RegistroSobreAltera.sobre_IOs_utilizados = Pines(0)
                                RegistroSobreAltera.sobre_IOs_disponibles = Pines(1)
                            End If

                            'Búsqueda de tiempos de CPU y la memoria utilizada
                            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                            '´Busca el tiempo de sintesis
                            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                            Dim AandS_time_columna As String
                            Dim AandS_time() As Integer
                            Dim memo_AandS_columna As String
                            Dim memo_AandS() As Integer
                      AandS_time_columna = busqueda_valor_flow(PathFicheroFlowAltera,_
                            "ANALYSIS & SYNTHESIS", 4)
                      AandS_time = tratar_columna_flow(AandS_time_columna, ":", True)
                            If AandS_time Is Nothing Then
                                AandS_time(0) = "0"
                                AandS_time(1) = "0"
                                AandS_time(2) = "0"
                            End If

                      memo_AandS_columna = busqueda_valor_flow(PathFicheroFlowAltera,_
                            "ANALYSIS & SYNTHESIS", 3)
                      memo_AandS = tratar_columna_flow(memo_AandS_columna, " ", True)

                            'Calcula el tiempo de sintesis en segundos
                            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                            Dim AandS_time_sec As Integer
                          AandS_time_sec = AandS_time(0) * 3600 + AandS_time(1) * 60 + AandS_time(2)

                            RegistroAltera.T_AandS = AandS_time_sec

                            '´Busca el tiempo de implementacion
                            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                            Dim fit_time_columna As String
                            Dim fit_time() As Integer
                            Dim memo_fit_columna As String
                            Dim memo_fit() As Integer
              fit_time_columna = busqueda_valor_flow(PathFicheroFlowAltera, "FITTER", 4)
                            fit_time = tratar_columna_flow(fit_time_columna, ":", True)
                            If fit_time Is Nothing Then
                                fit_time(0) = "0"
                                fit_time(1) = "0"
                                fit_time(2) = "0"
                            End If

              memo_fit_columna = busqueda_valor_flow(PathFicheroFlowAltera, "FITTER", 3)
                            memo_fit = tratar_columna_flow(memo_fit_columna, " ", True)

                            'Calcula el tiempo de fit en segundos
                            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                            Dim Fit_time_sec As Integer
              Fit_time_sec = fit_time(0) * 3600 + fit_time(1) * 60 + fit_time(2)

                            RegistroAltera.T_Fit = Fit_time_sec

                   'Calcula cuál ha sido la maxima memoria consumida entre los procesos
                            'Analysis o FIt
                            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                            Dim memo(1) As Integer
```

```
                          Dim pico_memo As Integer
                          memo(0) = memo_AandS(0)
                          memo(1) = memo_fit(0)
                          pico_memo = memo.Max

                          RegistroAltera.Pico_Memo = pico_memo
                      Else
                          '********
                          '* FALSO*
                          '********
                          'si no existe fichero, almacena de qué circuito se trata
             RegistroFalloAltera.fallo_flow = DirectoriosAlteraTecnologias.Item(k)

                      End If



                      '''''''''''''''''''
                      ' 2. POWER REPORT '
                      '''''''''''''''''''
                      'A la tecnologia se le añade la extension POW.SUMMARY

                      Dim NombreFicheroPotenciaAltera As String
                      Dim PathFicheroPotenciaAltera As String
                      Dim pot() As Single

                      NombreFicheroPotenciaAltera = circuito_altera + ".pow.summary"
            PathFicheroPotenciaAltera = DirectoriosAlteraTecnologias.Item(k) + "\" +_
                      NombreFicheroPotenciaAltera

'El siguiente bloque comprueba si el fichero POW.SUMMARY existe, busca la información
'necesaria si es TRUE y almacena el lugar de fallo si es FALSE
CompruebaFicheroExiste = My.Computer.FileSystem.FileExists(PathFicheroPotenciaAltera)

                      If CompruebaFicheroExiste = True Then
    'Se llama a la funcion de busqueda de informacion,ésta devuelve los dos valores
                          'existentes a ambos lados de un "/"
   pot = busqueda_valor(PathFicheroPotenciaAltera, "TOTAL THERMAL POWER DISSIPATION",_
                          ":", "", "MW")
                          RegistroAltera.potencia = pot(0)

                      Else
                          'si no existe fichero fit, almacena de qué circuito se trata
                          'e incrementa el contador de FIT fallidos
   RegistroFalloAltera.fallo_potencia = DirectoriosAlteraTecnologias.Item(k)
                          'RegistroAltera.potencia = 0
                          'DataGridView2.Rows.Add(fallos_pow(contador_fallos_pow - 1))
                      End If

                      '''''''''''''''''
                      ' 3. TIME REPORT '
                      '''''''''''''''''

                      'A la tecnologia se le añade la extension TAO
                      '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                      Dim P2P_time() As Single = {}
                      Dim Tsu_time() As Single
                      Dim Th_time() As Single
                      Dim Tco_time() As Single
                      Dim nombreFicheroTiempoAltera As String
                      Dim PathFicheroTiempoAltera As String
                      nombreFicheroTiempoAltera = circuito_altera + ".tao"
                      PathFicheroTiempoAltera = DirectoriosAlteraTecnologias.Item(k) +
                      "\" nombreFicheroTiempoAltera

  CompruebaFicheroExiste = My.Computer.FileSystem.FileExists(PathFicheroTiempoAltera)

                      If CompruebaFicheroExiste = True Then
 'Se llama a la funcion de busqueda de informacion,ésta devuelve los dos valores
 'existentes a ambos lados de un "/"
      P2P_time = busqueda_valor(PathFicheroTiempoAltera, "ACTUAL LONGEST P2P TIME", _
                          ":", "", "NS")
      Tsu_time = busqueda_valor(PathFicheroTiempoAltera, "ACTUAL TSU", ":", "", "NS")
                          RegistroAltera.Tsu = Tsu_time(0)
      Th_time = busqueda_valor(PathFicheroTiempoAltera, "ACTUAL TH", ":", "", "NS")
                          RegistroAltera.Th = Th_time(0)
      Tco_time = busqueda_valor(PathFicheroTiempoAltera, "ACTUAL TCO", ":", "", "NS")
                          RegistroAltera.Tco = Tco_time(0)
                      Else
                          'si no existe fichero fit, almacena de qué circuito se trata
                          'e incrementa el contador de FIT fallidos
          RegistroFalloAltera.fallo_tiempo = DirectoriosAlteraTecnologias.Item(k)
                          RegistroAltera.P2P = 0
                          RegistroAltera.Tsu = 0
                          RegistroAltera.Th = 0
```

```
                        RegistroAltera.Tco = 0
                    'DataGridView2.Rows.Add(fallos_time(contador_fallos_time - 1))
                      End If



                      '''''''''''''''''''''''''
                      ' 4. TIME REPORT(uTco y uTsu) '
                      '''''''''''''''''''''''''
                      'A la tecnologia se le añade la extension TAN.RPT
                      Dim NombreFicheroTANAltera As String
                      Dim PathFicheroTANAltera As String
                      Dim microtco_altera() As Single
                      Dim microtsu_altera() As Single
                      Dim SUMA As Single

                      NombreFicheroTANAltera = circuito_altera + ".tan.rpt"
        PathFicheroTANAltera = DirectoriosAlteraTecnologias.Item(k) + "\" + _
                      NombreFicheroTANAltera
        CompruebaFicheroExiste = My.Computer.FileSystem.FileExists(PathFicheroTANAltera)

                      If CompruebaFicheroExiste = True Then
         'Se llama a la funcion de busqueda de informacion,ésta devuelve los dos valores
         'existentes a ambos lados de un "/"
                       microtco_altera = busqueda_valor_xilinx(PathFicheroTANAltera, _
                          "INFO: + MICRO CLOCK TO OUTPUT DELAY OF SOURCE IS ")
                       microtsu_altera = busqueda_valor_xilinx(PathFicheroTANAltera, _
                          "INFO: + MICRO SETUP DELAY OF DESTINATION IS ")
                          SUMA = P2P_time(0) + microtco_altera(0) + microtsu_altera(0)
                          RegistroAltera.P2P = SUMA
                      Else
                          'si no existe fichero tan, almacena de qué circuito se trata
                          'e incrementa el contador de tan fallidos
                RegistroFalloAltera.fallo_tiempo = DirectoriosAlteraTecnologias.Item(k)
                          RegistroAltera.P2P = 0
                          RegistroAltera.Tsu = 0
                          RegistroAltera.Th = 0
                          RegistroAltera.Tco = 0
                      End If

        '-----------------------------------------------------------------------------
        'Actualización de la base de datos de access. Estas lineas son las que realmente
        'añaden los registros a las tablas de access
        '-----------------------------------------------------------------------------
          'Los resultados leidos se añaden a la tabla DatosReports en el caso de que
          'no haya habido sobredimensionado por UL o pines. Tampoco irán a esta tabla si
          'se ha presentado algún fallo en los ficheros
           '-------------------------------------------------------------------
                      '¿Hay sobredimensionado?
                      '''''''''''''''''''''''''''''''''''
                      If RegistroSobreAltera.sobre_IOs_utilizados <> 0 Or_
                      RegistroSobreAltera.sobre_ULs_utilizados <> 0 Then
                          'SI
                    RegistroSobreAltera.sobre_circuito = RegistroAltera.Circuito
                       RegistroSobreAltera.sobre_fabricante = RegistroAltera.Fabricante
                    RegistroSobreAltera.sobre_tecnologia = RegistroAltera.Tecnologia

SobredimensionadosTableAdapter.Insert(RegistroSobreAltera.sobre_fabricante, _
RegistroSobreAltera.sobre_circuito, _
RegistroSobreAltera.sobre_estrategia, _
RegistroSobreAltera.sobre_tecnologia, _
RegistroSobreAltera.sobre_sector, _
RegistroSobreAltera.sobre_ULs_utilizados, _
RegistroSobreAltera.sobre_ULs_disponibles, _
RegistroSobreAltera.sobre_IOs_utilizados, _
RegistroSobreAltera.sobre_IOs_disponibles)
                          'NO
                          'Hay fallos (falta algún fichero)?
                      ElseIf RegistroFalloAltera.fallo_implementacion <> "" Or_
                      RegistroFalloAltera.fallo_potencia <> "" _
                                      Or RegistroFalloAltera.fallo_tiempo <> "" _
                                       Or RegistroFalloAltera.fallo_flow <> "" Then
                          'SI
                          FallosTableAdapter.Insert(RegistroFalloAltera.fallo_Circuito, _
RegistroFalloAltera.fallo_Estrategias, _
RegistroFalloAltera.fallo_Tecnologia, _
RegistroFalloAltera.fallo_implementacion, _
RegistroFalloAltera.fallo_potencia, _
RegistroFalloAltera.fallo_tiempo, _
RegistroFalloAltera.fallo_flow)
                          'NO
                      Else
                          'NO
   DatosReportsTableAdapter.Insert(RegistroAltera.Fabricante, RegistroAltera.Circuito,_
RegistroAltera.Estrategias, RegistroAltera.Tecnologia, RegistroAltera.Sector,_
```

```
         RegistroAltera.UL_usados, RegistroAltera.UL_existentes,RegistroAltera.Bits_usados,_
RegistroAltera.Bits_existentes, RegistroAltera.Mult_9_usados,_
RegistroAltera.Mult_9_existentes, RegistroAltera.Pines_usados, _
RegistroAltera.Pines_existentes, RegistroAltera.potencia, RegistroAltera.P2P, _
RegistroAltera.Tsu, RegistroAltera.Th, RegistroAltera.Tco, RegistroAltera.T_AandS, _
                            RegistroAltera.T_Fit, RegistroAltera.Pico_Memo)
                        End If


                  'Se resetean todos los valores por si acaso el próximo circuito leido
                  'no dispone de algun fichero. Si no se hace se duplican los registros.
                  'Se incluyen en la tabla principal con el valor del circuito anterio
                        '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                        RegistroFalloAltera.fallo_implementacion = ""
                        RegistroFalloAltera.fallo_potencia = ""
                        RegistroFalloAltera.fallo_tiempo = ""
                        RegistroFalloAltera.fallo_flow = ""


                        RegistroSobreAltera.sobre_circuito = ""
                        RegistroSobreAltera.sobre_fabricante = ""
                        RegistroSobreAltera.sobre_tecnologia = ""
                        RegistroSobreAltera.sobre_ULs_disponibles = 0
                        RegistroSobreAltera.sobre_ULs_utilizados = 0
                        RegistroSobreAltera.sobre_IOs_disponibles = 0
                        RegistroSobreAltera.sobre_IOs_utilizados = 0

                        RegistroAltera.Bits_existentes = 0
                        RegistroAltera.Bits_usados = 0
                        RegistroAltera.Circuito = ""
                        RegistroAltera.Estrategias = ""
                        RegistroAltera.Fabricante = ""
                        RegistroAltera.Mult_9_existentes = 0
                        RegistroAltera.Mult_9_usados = 0
                        RegistroAltera.P2P = 0
                        RegistroAltera.Pico_Memo = 0
                        RegistroAltera.Pines_existentes = 0
                        RegistroAltera.Pines_usados = 0
                        RegistroAltera.potencia = 0
                        RegistroAltera.Sector = ""
                        RegistroAltera.T_AandS = 0
                        RegistroAltera.T_Fit = 0
                        RegistroAltera.Tco = 0
                        RegistroAltera.Tecnologia = ""
                        RegistroAltera.Th = 0
                        RegistroAltera.Tsu = 0
                        '---------------------------------------------------------------------------
--
                    Next
                Next
            Next

            TextBox2.Text = DatosReportsBindingSource.Count


            '____________________   XILINX     _____________________
            '

            Dim DirectoriosXilinx As System.Collections.ObjectModel.ReadOnlyCollection(Of String)
            Dim NumDirXilinx As Integer
            Dim DirectoriosXilinxCircuitos As System.Collections.ObjectModel.ReadOnlyCollection(Of
String)
            Dim NumDirXilinxCircuitos As Integer
            Dim DirectoriosXilinxTecnologias As System.Collections.ObjectModel.ReadOnlyCollection(Of
String)
            Dim NumDirXilinxTecnologias As Integer
            Dim sector_Xilinx As String
            Dim fabricante_Xilinx As String
            Dim estrategia_Xilinx As String
            Dim circuito_Xilinx As String
            Dim tecnologia_Xilinx As String
            Dim RegistroXilinx As New Registro
            Dim RegistroFalloXilinx As New Fallo
            Dim RegistroSobreXilinx As New Sobredimension
            Dim NombreFicheroRecursosXilinx As String
            Dim PathFicheroRecursosXilinx As String



  'Obtiene el nombre de la carpeta del siguiente nivel(Fabricante -Estrategia - Sector)
            'En principio deben ser 6
            DirectoriosXilinx = My.Computer.FileSystem.GetDirectories(ruta_Xilinx)
            NumDirXilinx = DirectoriosXilinx.Count

  'Obtiene el nombre de todas las carpetas dentro de cada una de las anteriores
            'Nivel Circuitos (por cada fabricante-estrategia-sector)
```

```
            'En principio 15 en total por cada estrategia
            For i = 0 To NumDirXilinx - 1
                DirectoriosXilinxCircuitos = My.Computer.FileSystem.GetDirectories_
                (DirectoriosXilinx.Item(i))
                NumDirXilinxCircuitos = DirectoriosXilinxCircuitos.Count
                'Nivel Tecnologias por Circuitos
                For j = 0 To NumDirXilinxCircuitos - 1
                   DirectoriosXilinxTecnologias = My.Computer.FileSystem.GetDirectories_
                    (DirectoriosXilinxCircuitos.Item(j))
                    NumDirXilinxTecnologias = DirectoriosXilinxTecnologias.Count
                    'Nivel ficheros resultantes a leer
                    For k = 0 To NumDirXilinxTecnologias - 1

            'Determinación de Fabricante,Circuito, EStrategia,FPGA,Sector(ese orden)
            '---------------------------------------------------------------------------
                        Dim ver_datos_xilinx() As String
                        ver_datos_xilinx = datos(DirectoriosXilinxTecnologias(k))

                        sector_Xilinx = ver_datos_xilinx(0)
                        fabricante_Xilinx = ver_datos_xilinx(1)
                        estrategia_Xilinx = ver_datos_xilinx(2)
                        circuito_Xilinx = ver_datos_xilinx(3)
                        tecnologia_Xilinx = ver_datos_xilinx(4)


                        'Se almacena en la estructura creada el valor de cada campo
                        RegistroXilinx.Fabricante = fabricante_Xilinx.ToUpper
                        RegistroXilinx.Circuito = circuito_Xilinx.ToUpper
                        RegistroXilinx.Estrategias = estrategia_Xilinx.ToUpper
                        RegistroXilinx.Tecnologia = tecnologia_Xilinx
                        RegistroXilinx.Sector = sector_Xilinx.ToUpper


                        RegistroFalloXilinx.fallo_Circuito = circuito_Xilinx.ToUpper
                       RegistroFalloXilinx.fallo_Estrategias = estrategia_Xilinx.ToUpper
                        RegistroFalloXilinx.fallo_Tecnologia = tecnologia_Xilinx




            '---------------------------------------------------------------------------
            'Abrir ficheros de resultados
            '---------------------------------------------------------------------------
'El nombre de los ficheros coincide con el del circuito que se está analizando y que
'está almacenado en circuito_altera. En cada caso se le añade la extensión adecuada.

                        '''''''''''''''''
                        ' 1. FICHERO PAR'
                        '''''''''''''''''
                        Dim memo_par(1) As Single
                        Dim par_time(1) As Single
                        Dim par_time_sec As Single
                        Dim LEs() As Single
                        Dim pines() As Single
                        Dim bits() As Single
                        Dim mult() As Single
                        'Al circuito se le añade la extension PAR
                        NombreFicheroRecursosXilinx = circuito_Xilinx + ".par"
                PathFicheroRecursosXilinx = DirectoriosXilinxTecnologias.Item(k) + "\" _
                        + NombreFicheroRecursosXilinx
          'El siguiente bloque comprueba si el fichero PAR existe, busca la información
          'necesaria si es TRUE y almacena el lugar de fallo si es FALSE
                        '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
  CompruebaFicheroExiste = My.Computer.FileSystem.FileExists(PathFicheroRecursosXilinx)

                        '´´´´´´´´´´´´
                        ' VERDADERO
                        '´´´´´´´´´´´´


                        If CompruebaFicheroExiste = True Then

                            '´´´´´´´´
                            ' AREA
                            '´´´´´´´´
    LEs = busqueda_valor_xilinx(PathFicheroRecursosXilinx, "NUMBER OF SLICES")
    pines = busqueda_valor_xilinx(PathFicheroRecursosXilinx, "NUMBER OF EXTERNAL IOBS")
           'Comprobación de que no hay sobredimensionado de recursos: slices o pines
                            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                            If (LEs(0) < LEs(1) And pines(0) < pines(1)) Then

                                'Continua leyendo si no hay overmapping
                                '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´


       bits = busqueda_valor_xilinx(PathFicheroRecursosXilinx, "NUMBER OF RAMB16S")

                                'Busca el resto de formato de los bloques de memoria
```

```
                                'Difiere en algunas tecnologías
                                '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
if (bits(0) = 0 And bits(1) = 0) Then
                                    bits=busqueda_valor_xilinx(PathFicheroRecursosXilinx, _
                                    "NUMBER OF RAMB16BWES")
                                End If

        'Xilinx expresa en su informe el numero de bloques empleados, se ha optado
        'por utilizar el formato de Altera, memoria expresada en bits. Se multiplica por
        'el total de bits que supone cada bloque memo de xilinx

'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                                bits(0) = bits(0) * 18432
                                bits(1) = bits(1) * 18432

 mult = busqueda_valor_xilinx(PathFicheroRecursosXilinx, "NUMBER OF MULT18X18SIOS")

 'Al igual que las primitivas de memoria, Xilinx utiliza varias de multiplicadores
 'segun la tecnologia de que se trate. Se buscan las dos formas en los report
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                                If (mult(0) = 0 And mult(1) = 0) Then
    mult = busqueda_valor_xilinx(PathFicheroRecursosXilinx, "NUMBER OF MULT18X18S")
                                End If

                                'Tiempo utilizado para el proceso PAR
                                '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
    par_time = busqueda_valor_xilinx(PathFicheroRecursosXilinx,_
                                "TOTAL CPU TIME TO PAR COMPLETION:")

                                'Conversión a segundos del tiempo completo

'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                                par_time_sec = par_time(0) * 60
                                par_time_sec = par_time_sec + par_time(1)

                                'Memoria utilizada para el proceso PAR
                                '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
   memo_par = busqueda_valor_xilinx(PathFicheroRecursosXilinx, "PEAK MEMORY USAGE:")

                                'Actualizacion de los registros
                                '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                                RegistroXilinx.UL_usados = LEs(0)
                                RegistroXilinx.UL_existentes = LEs(1)
                                RegistroXilinx.Pines_usados = pines(0)
                                RegistroXilinx.Pines_existentes = pines(1)
                                RegistroXilinx.Bits_usados = bits(0)
                                RegistroXilinx.Bits_existentes = bits(1)
                                RegistroXilinx.Mult_9_usados = mult(0)
                                RegistroXilinx.Mult_9_existentes = mult(1)
                                RegistroXilinx.T_Fit = par_time_sec
                                RegistroXilinx.Pico_Memo = memo_par(0)
                            Else
                                'Caso de que haya overmapping por Slices o pines
                                '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                                RegistroSobreXilinx.sobre_IOs_disponibles = pines(1)
                                RegistroSobreXilinx.sobre_IOs_utilizados = pines(0)
                                RegistroSobreXilinx.sobre_ULs_utilizados = LEs(0)
                                RegistroSobreXilinx.sobre_ULs_disponibles = LEs(1)
                            End If

                        Else

                            '`´´´´´´´´
                            ' FALSO  ´
                            '´´´´´´´´´
                            'Si no existe fichero PAR, almacena de qué circuito se trata
                            'e incrementa el contador de FIT fallidos
                            'par_time_sec = 0
                            'memo_par(0) = 0
                       'se añade para que tenga valor nulo, ya que se necsita más tarde
                            'y si no pasa por el bloque de antes no toma valor

     RegistroFalloXilinx.fallo_implementacion = DirectoriosXilinxTecnologias.Item(k)
                        End If

                        ''''''''''''''''''''''''''''
                        ' 2. FICHERO SRP (síntesis)'
                        ''''''''''''''''''''''''''''
                        ' Se obtiene información sobre tiempo de CPU y memoria empleada
                        ' Tambien puede ser que haya creado un fichero LOG
                        '---------------------------------------------------------------
                        Dim NombreFicheroSRPXilinx As String
                        Dim NombreFicheroLOGXilinx As String
                        Dim PathFicheroSRPXilinx As String
                        Dim PathFicheroLOGXilinx As String
                        Dim srp_time(1) As Single
```

```
                        Dim memo_srp(1) As Single
                        Dim memo_srp_MB As Single
                        Dim CompruebaFicheroExiste1 As Boolean
                        Dim srp_time_sec As Single


                        NombreFicheroSRPXilinx = circuito_Xilinx + ".srp"
                        NombreFicheroLOGXilinx = circuito_Xilinx + "_xst.log"
                     PathFicheroSRPXilinx = DirectoriosXilinxTecnologias.Item(k) + "\"_
                        + NombreFicheroSRPXilinx
                     PathFicheroLOGXilinx = DirectoriosXilinxTecnologias.Item(k) + "\"_
                        + NombreFicheroLOGXilinx

       'El siguiente bloque comprueba si el fichero SRP existe, busca la información
      'necesaria si es TRUE y almacena el lugar de fallo si es FALSE

'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
    CompruebaFicheroExiste = My.Computer.FileSystem.FileExists(PathFicheroSRPXilinx)
    CompruebaFicheroExiste1 = My.Computer.FileSystem.FileExists(PathFicheroLOGXilinx)


                        If CompruebaFicheroExiste = True Then

                            '''''''''''''
                            ' VERDADERO '
                            '''''''''''''

      'En este informe aparecen tres tiempos de CPU, ya que tambien notifica los de la
      'configuracion del directorio de trabajo y del temporal
      'Se hace necesaria una nueva función que no salga del fichero una vez localizada
      'la cadena que se le pasa.
                            srp_time = busqueda_ultimavez(PathFicheroSRPXilinx, _
                            "TOTAL CPU TIME TO XST COMPLETION:")

                            'Conversión a segundos del tiempo completo
                            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                            srp_time_sec = srp_time(0) * 60
                            srp_time_sec = srp_time_sec + srp_time(1)

                            RegistroXilinx.T_AandS = srp_time_sec

        memo_srp = busqueda_valor_xilinx(PathFicheroSRPXilinx, "TOTAL MEMORY USAGE IS")
        memo_srp_MB = memo_srp(0) / 1024

                        ElseIf CompruebaFicheroExiste1 = True Then
                            'Comprueba si hay fichero LOG
                            'Busqueda de tiempo de CPU
                            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                            srp_time = busqueda_ultimavez(PathFicheroLOGXilinx,_
                            "TOTAL CPU TIME TO XST COMPLETION:")

                            'Conversión a segundos del tiempo completo
                            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                            srp_time_sec = srp_time(0) * 60
                            srp_time_sec = srp_time_sec + srp_time(1)

                            RegistroXilinx.T_AandS = srp_time_sec

                            'Busqueda de la memoria utilizada
                            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
        memo_srp = busqueda_valor_xilinx(PathFicheroLOGXilinx, "TOTAL MEMORY USAGE IS")
                            memo_srp_MB = memo_srp(0) / 1024
                        Else
                    'si no existe fichero SRP ni LOG, almacena de qué circuito se trata
                    'e incrementa el contador de SRP fallidos
                            memo_srp_MB = 0
            RegistroFalloXilinx.fallo_flow = DirectoriosXilinxTecnologias.Item(k)
                        End If

                        ''''''''''''''''''''''''''''
                        ' 3. FICHERO MAP (mapping)'
                        ''''''''''''''''''''''''''''
        ' Se obtiene información sobre tiempo de CPU y memoria empleada en el mapping
        '------------------------------------------------------------------------------
                        Dim NombreFicheroMAPXilinx As String
                        Dim PathFicheroMAPXilinx As String
                        Dim map_time(1) As Single
                        Dim map_time_sec As Single
                        Dim memo_map(1) As Single


                        NombreFicheroMAPXilinx = circuito_Xilinx + "_map.map"
                   PathFicheroMAPXilinx = DirectoriosXilinxTecnologias.Item(k) + "\" +_
                        NombreFicheroMAPXilinx

        'El siguiente bloque comprueba si el fichero SRP existe, busca la información
```

```
                        'necesaria si es TRUE y almacena el lugar de fallo si es FALSE
      CompruebaFicheroExiste = My.Computer.FileSystem.FileExists(PathFicheroMAPXilinx)

                        If CompruebaFicheroExiste = True Then
     'Busqueda del tiempo de CPU del proceso de mapping.Se sumará a la de PAR para tener el tiempo
total de implementacion
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                            map_time = busqueda_ultimavez(PathFicheroMAPXilinx,_
                            "TOTAL CPU TIME TO MAP COMPLETION:")
                            map_time_sec = map_time(0) * 60 + map_time(1)

                            'Busqueda de la memoria de pico.
                            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
        memo_map = busqueda_valor_xilinx(PathFicheroMAPXilinx, "PEAK MEMORY USAGE:")

                        Else
                            'si no existe fichero MAP, almacena de qué circuito se trata
                            'e incrementa el contador de FIT fallidos
                            memo_map(0) = 0
     RegistroFalloXilinx.fallo_implementacion = DirectoriosXilinxTecnologias.Item(k)
                        End If


                        ''''''''''''''''''''''''''''
                        ' 4. FICHERO PWR (potencia)'
                        ''''''''''''''''''''''''''''
        ' Se obtiene información sobre el consumo dinámico sumado al estático en mW
        '------------------------------------------------------------------------------
                        Dim NombreFicheroPWRXilinx As String
                        Dim PathFicheroPWRXilinx As String
                        Dim pwr_xilinx As String

                        NombreFicheroPWRXilinx = circuito_Xilinx + ".pwr"
              PathFicheroPWRXilinx = DirectoriosXilinxTecnologias.Item(k) + "\" + _
                        NombreFicheroPWRXilinx

      'El siguiente bloque comprueba si el fichero PWR existe, busca la información
      'necesaria si es TRUE y almacena el lugar de fallo si es FALSE
       CompruebaFicheroExiste = My.Computer.FileSystem.FileExists(PathFicheroPWRXilinx)

                        If CompruebaFicheroExiste = True Then
                            'Busqueda de la potencia de consumo
                            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                            pwr_xilinx = busqueda_valor_separa(PathFicheroPWRXilinx, _
                            "TOTAL ESTIMATED POWER CONSUMPTION", "|", 2)
                            RegistroXilinx.potencia = pwr_xilinx
                        Else
                            'si no existe fichero PWR, almacena de qué circuito se trata
       RegistroFalloXilinx.fallo_potencia = DirectoriosXilinxTecnologias.Item(k)
                        End If


                        ''''''''''''''''''''''''''''
                        ' 5. FICHERO TWR (tiempos)'
                        ''''''''''''''''''''''''''''
       ' Se obtiene información sobre setup, hold, clock-to-out y periodo máximo
       '------------------------------------------------------------------------------

                        Dim NombreFicheroTWRXilinx As String
                        Dim PathFicheroTWRXilinx As String
                        Dim tsu_xilinx() As String
                        Dim th_xilinx() As String
                        Dim tco_xilinx() As String
                        Dim clock_xilinx() As String
                        Dim tsu_single() As Single
                        Dim th_single() As Single
                        Dim tsu_max As String
                        Dim th_max As String
                        Dim tco_max As String
                        Dim num_elem_col As Integer
                        Dim clock_single As Single
                        Dim memo_twr(1) As Single


                        NombreFicheroTWRXilinx = circuito_Xilinx + ".twr"
              PathFicheroTWRXilinx = DirectoriosXilinxTecnologias.Item(k) + "\" + _
                        NombreFicheroTWRXilinx

        'El siguiente bloque comprueba si el fichero TWR existe, busca la información
        'necesaria si es TRUE y almacena el lugar de fallo si es FALSE
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
CompruebaFicheroExiste = My.Computer.FileSystem.FileExists(PathFicheroTWRXilinx)

                        If CompruebaFicheroExiste = True Then
                            '''''''''''''
                            ' VERDADERO '
```

```
                            '''''''''''''

      'Busqueda de los parámetros de tiempo. xilinx ordena por nombre de net y no por
      'valor del retraso. Eso supone tener que leer todos los retardos de todas las net
      'y averiguar cual es el más desfavorable
      '--------------------------------------------------------------------------------
          tsu_xilinx = leer_columna_tabla(PathFicheroTWRXilinx, "SETUP/HOLD TO CLOCK",_
                            5, "|", 1)
          th_xilinx = leer_columna_tabla(PathFicheroTWRXilinx, "SETUP/HOLD TO CLOCK", _
                            5, "|", 2)
           tco_xilinx = leer_columna_tabla(PathFicheroTWRXilinx, "DESTINATION", 2,_
                            "|", 1)
          clock_xilinx = leer_columna_tabla(PathFicheroTWRXilinx,_
                            "CLOCK TO SETUP ON DESTINATION CLOCK", 5, "|", 1)

                    'Redimensiona los array, a priori de longitud desconocida ya que
                    'depende del total de nets de cada circuito.
                    'Ahora ya se puede determinar el total de elemento de que dispone
                    'Se recorre todo el array y se modifican los valores para que queden
                    'solo numero single y notacion decimal con ","
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                            num_elem_col = tsu_xilinx.Length - 1

                            ReDim Preserve tsu_single(num_elem_col)
                            ReDim Preserve th_single(num_elem_col)

                        'Recorta el elemento (quita el tipo de flanco especificado(R)).
                        'Cambia el punto decimal por coma decimal para Tsu y Th
                        'Convierte a simple precision.
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                            'tsu y th'
                            ''''''''''
                            For i_recorte = 0 To num_elem_col
                  tsu_xilinx(i_recorte) = elimina_trozo(tsu_xilinx(i_recorte), "(", ")")
                  tsu_xilinx(i_recorte) = tsu_xilinx(i_recorte).Replace(".", ",")
                  tsu_single(i_recorte) = tsu_xilinx(i_recorte)
                  th_xilinx(i_recorte) = elimina_trozo(th_xilinx(i_recorte), "(", ")")
                  th_xilinx(i_recorte) = th_xilinx(i_recorte).Replace(".", ",")
                  th_single(i_recorte) = th_xilinx(i_recorte)

                'Otra manera de hacer la conversion string->single
                'th_single(i_recorte) = Single.Parse(th_xilinx(i_recorte))
                            Next

                            '''''''
                            ' tco '
                            '''''''
                            For i_recorte = 0 To tco_xilinx.Length - 1
              tco_xilinx(i_recorte) = elimina_trozo(tco_xilinx(i_recorte), "(", ")")
              tco_xilinx(i_recorte) = tco_xilinx(i_recorte).Replace(".", ",")
                            Next

                            '´''''''''''''''''''''''''''''''''''''
                            ' Clock to setup on destination clock '
                            ''''''''''''''''''''''''''''''''''''''
                            clock_xilinx(0) = clock_xilinx(0).Replace(".", ",")
                            clock_single = clock_xilinx(0)

                           'Con los datos ya como numeros, extrae el de mayor valor para
                           'cada parametro.(Tsu,Th,Tco)
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                            If tsu_single.Length > 0 Then
                                tsu_max = tsu_single.Max
                            Else
                                tsu_max = 0
                            End If

                            If th_single.Length > 0 Then
                                th_max = th_single.Max
                            Else
                                th_max = 0
                            End If

                            If tco_xilinx.Length > 0 Then
                                tco_max = tco_xilinx.Max
                            Else
                                tco_max = 0
                            End If

                            'Asignacion a registros
                            '´´´´´´´´´´´´´´´´´´´´´´
                            RegistroXilinx.Tsu = tsu_max
                            RegistroXilinx.Th = th_max
                            RegistroXilinx.Tco = tco_max
                            RegistroXilinx.P2P = clock_single
```

```
                        Else
                            'si no existe fichero PWR, almacena de qué circuito se trata
                RegistroFalloXilinx.fallo_tiempo = DirectoriosXilinxTecnologias.Item(k)
                        End If


                        '-------------------------------------------------------
                        'Cálculo de la máxima memoria de pico empleado
                        'Máximo de todas o bien solo la de A&S si no ha habido FIT
                        '-------------------------------------------------------
                        Dim memo_xilinx(2) As Single
                        Dim pico_memo_xilinx As Single

      'De memoria se devuelve el mayor pico que ha demandado alguna de las aplicaciones
                        'Se limpian los espacios de principio y fin de la cadena
                        memo_xilinx(0) = memo_srp_MB
                        memo_xilinx(1) = memo_map(0)
                        memo_xilinx(2) = memo_par(0)
                        pico_memo_xilinx = memo_xilinx.Max

                        RegistroXilinx.Pico_Memo = pico_memo_xilinx

       '-----------------------------------------------------------------------------
       'Actualización de la base de datos de access
       '-----------------------------------------------------------------------------
                        '¿Hay sobredimensionados?
                        If RegistroSobreXilinx.sobre_IOs_utilizados <> 0 Or_
                        RegistroSobreXilinx.sobre_ULs_utilizados <> 0 Then
                            'SI
        RegistroSobreXilinx.sobre_circuito = RegistroXilinx.Circuito
       RegistroSobreXilinx.sobre_fabricante = RegistroXilinx.Fabricante
       RegistroSobreXilinx.sobre_tecnologia = RegistroXilinx.Tecnologia
       RegistroSobreXilinx.sobre_estrategia = RegistroXilinx.Estrategias

SobredimensionadosTableAdapter.Insert(RegistroSobreXilinx.sobre_fabricante, _
RegistroSobreXilinx.sobre_circuito, _
RegistroSobreXilinx.sobre_estrategia, _
RegistroSobreXilinx.sobre_tecnologia, _
RegistroSobreXilinx.sobre_sector, _
RegistroSobreXilinx.sobre_ULs_utilizados, _
RegistroSobreXilinx.sobre_ULs_disponibles, _
RegistroSobreXilinx.sobre_IOs_utilizados, _
RegistroSobreXilinx.sobre_IOs_disponibles)
                            'NO
                            '¿Hay ficheros fallidos?
                        ElseIf RegistroFalloXilinx.fallo_implementacion <> "" Or_
                         RegistroFalloXilinx.fallo_potencia <> "" Or
                         RegistroFalloXilinx.fallo_tiempo_
                        <> "" Or RegistroFalloXilinx.fallo_flow <> "" Then
                            'SI
                                 FallosTableAdapter.Insert(RegistroFalloXilinx.fallo_Circuito,
        RegistroFalloXilinx.fallo_Estrategias, RegistroFalloXilinx.fallo_Tecnologia,
        RegistroFalloXilinx.fallo_implementacion, RegistroFalloXilinx.fallo_potencia,
        RegistroFalloXilinx.fallo_tiempo, RegistroFalloXilinx.fallo_flow)
                        Else
                            'NO
DatosReportsTableAdapter.Insert(RegistroXilinx.Fabricante, RegistroXilinx.Circuito,_
   RegistroXilinx.Estrategias, RegistroXilinx.Tecnologia, RegistroXilinx.Sector,_
RegistroXilinx.UL_usados, RegistroXilinx.UL_existentes, RegistroXilinx.Bits_usados,_
RegistroXilinx.Bits_existentes, RegistroXilinx.Mult_9_usados,
RegistroXilinx.Mult_9_existentes, RegistroXilinx.Pines_usados,
RegistroXilinx.Pines_existentes, RegistroXilinx.potencia, RegistroXilinx.P2P,
RegistroXilinx.Tsu, RegistroXilinx.Th, RegistroXilinx.Tco, RegistroXilinx.T_AandS,
RegistroXilinx.T_Fit, RegistroXilinx.Pico_Memo)
                        End If


                        'se reinician todos los registros de esta iteración
                        RegistroFalloXilinx.fallo_implementacion = ""
                        RegistroFalloXilinx.fallo_potencia = ""
                        RegistroFalloXilinx.fallo_tiempo = ""
                        RegistroFalloXilinx.fallo_flow = ""

                        RegistroSobreXilinx.sobre_IOs_utilizados = 0
                        RegistroSobreXilinx.sobre_IOs_disponibles = 0
                        RegistroSobreXilinx.sobre_ULs_utilizados = 0
                        RegistroSobreXilinx.sobre_ULs_disponibles = 0

                        RegistroXilinx.Fabricante = ""
                        RegistroXilinx.Circuito = ""
                        RegistroXilinx.Estrategias = ""
                        RegistroXilinx.Tecnologia = ""
                        RegistroXilinx.Sector = ""
                        RegistroXilinx.UL_usados = 0
                        RegistroXilinx.UL_existentes = 0
```

```
                        RegistroXilinx.Pines_usados = 0
                        RegistroXilinx.Pines_existentes = 0
                        RegistroXilinx.Bits_usados = 0
                        RegistroXilinx.Bits_existentes = 0
                        RegistroXilinx.Mult_9_usados = 0
                        RegistroXilinx.Mult_9_existentes = 0
                        RegistroXilinx.potencia = 0
                        RegistroXilinx.P2P = 0
                        RegistroXilinx.Tsu = 0
                        RegistroXilinx.Th = 0
                        RegistroXilinx.Tco = 0
                        RegistroXilinx.T_AandS = 0
                        RegistroXilinx.T_Fit = 0
                        RegistroXilinx.Pico_Memo = 0
         '-----------------------------------------------------------------------------
                    Next
                Next
            Next

        Else
            TextBox_ruta.Text = ""
            MsgBox("El directorio seleccionado no es correcto")

        End If

        MsgBox("El proceso ha finalizado")

        Me.FallosTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Fallos)
        Me.DatosReportsTableAdapter.Fill(Me.Resultados_benchamrkDataSet.DatosReports)
     Me.SobredimensionadosTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Sobredimensionados)
        TextBox2.Text = DataGridView1.RowCount - 1
        TextBox3.Text = FallosDataGridView.RowCount - 1
        TextBox4.Text = SobredimensionadosDataGridView.RowCount - 1

    End Sub

    Private Function datos(ByVal RutaCompleta As String)
        Dim long_ruta As Integer
        Dim array_ruta() As String
        Dim array_datos(4) As String
        Dim s_f_e As String
        Dim array_s_f_e(2) As String

        array_ruta = RutaCompleta.Split("\")
        long_ruta = array_ruta.Length
        s_f_e = array_ruta(long_ruta - 3)
        array_s_f_e = s_f_e.Split("_")

        array_datos(0) = array_s_f_e(0)
        array_datos(1) = array_s_f_e(1)
        array_datos(2) = array_s_f_e(2)
        array_datos(3) = array_ruta(long_ruta - 2)
        array_datos(4) = array_ruta(long_ruta - 1)

        Return array_datos

    End Function

    Private Function busqueda_valor(ByVal camino_fileread As String, ByVal palabra As String, ByVal
car_inicio As Char, ByVal car_medio As String, ByVal car_fin As String)
        '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
        'Devuelve los valores existentes entre los caracteres indicados
        'Sirve si los caracteres aparecen una única vez
''´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
        Dim fichero As IO.StreamReader
        Dim linea As String
        Dim ind_inicio As Integer
        Dim valores(1) As Single
        Dim valor_disponible As Single
        Dim valor_usado As Single
        Dim ind_medio As Integer
        Dim ind_fin As Integer
        Dim longitud_disponible As Integer
        Dim longitud_usado As Integer
        Dim bandera_InStr As Boolean = False

        'Se asigna a la variable "fichero" el fichero que debe abrir
        fichero = IO.File.OpenText(camino_fileread)
        'Se asigna a la variable "linea" una linea del fichero que se abre
        'para ir localalizando el texto deseado
        linea = fichero.ReadLine

        Resultados_benchamrkDataSet.DatosReports.Clear()
        Resultados_benchamrkDataSet.AcceptChanges()
        'Mientra no se llegue al final del fichero
        While Not linea Is Nothing And bandera_InStr = False
```

```
            'Convierte la linea a mayusculas
            linea = linea.ToUpper
            'elimina los espacios del inicio y el fin de la linea, asi como el TAB
            linea = linea.Trim(" ", Chr(9))
            'sustituye la , de millares por nada
            linea = linea.Replace(",", "")
            'sustituye el . decimal por ,
            linea = linea.Replace(".", ",")
            'Continua el proceso si la linea no es un espacio
            If Not linea = "" Then
                If InStr(linea, palabra) Then

                    'bandera de aviso de que ha encontrado ya el texto deseado
                    bandera_InStr = True

                    If linea.StartsWith(";") Then
                        linea = linea.Substring(1)
                    End If

                    'determina la posicion en la línea donde esta car_inicio
                    ind_inicio = linea.IndexOf(car_inicio)

                    'determina la posicion en la línea de car_medio
                    ind_medio = linea.IndexOf(car_medio)

                    'determina la posicion en car_fin
                    ind_fin = linea.IndexOf(car_fin)
                    If ind_medio <> 0 Then
         'lee todo lo que hay entre car_inicio y car_medio  y entre car_medio y car_fin
                longitud_usado = ind_medio - (ind_inicio + 1)
                valor_usado = linea.Substring(ind_inicio + 1, longitud_usado)
                longitud_disponible = ind_fin - (ind_medio + 1)
                valor_disponible = linea.Substring(ind_medio + 1, longitud_disponible)
                        valores(0) = valor_usado
                        valores(1) = valor_disponible
                    Else
                        longitud_usado = ind_fin - (ind_inicio + 1)
                        valor_usado = linea.Substring(ind_inicio + 1, longitud_usado)
                        valor_disponible = 0
                        valores(0) = valor_usado
                    End If
                End If
            End If

            'lee la siguiente linea
            linea = fichero.ReadLine

        End While

        'cierra el fichero una vez leido entero
        fichero.Close()
        Return valores
    End Function

    Private Function elimina_trozo(ByVal elemento As String, ByVal car_inicio As Char, ByVal car_fin
As Char)
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        Dim elemento_recortado As String
        Dim ind_inicio As Integer
        Dim ind_fin As Integer

        ind_inicio = elemento.IndexOf(car_inicio)
        ind_fin = elemento.IndexOf(car_fin)
        elemento_recortado = elemento.Remove(ind_inicio, (ind_fin - ind_inicio) + 1)

        Return elemento_recortado

    End Function

    Private Function leer_columna_tabla(ByVal camino_fileread As String, ByVal palabra As String,
ByVal inicio_lectura As Integer, ByVal car_separador_columna As Char, ByVal num_columna As Integer)
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Localiza la linea del fichero que contiene un determinado.
' A continuación, se posiciona en la linea actual + "inicio_lectura". Forma un array con los
elementos de la nueva linea separados por "car_separador_columna". El elemento de este array situando
en "num_columna", se guarda en otro (columna de la tabla)
' Crea otro array con los elementos  "num_columna"
' Repite el proceso hasta que encuentre una linea que no tenga el caracter separador
'---------------------------------------------------------------------------------------
        Dim fichero As IO.StreamReader
        Dim linea As String
        Dim bandera_InStr As Boolean = False
        Dim array_fila() As String
        Dim array_columna() As String = {}
```

```
        Dim total_filas As Integer
        Dim columna() As String = {}


        'Se asigna a la variable "fichero" el fichero que debe abrir
        fichero = IO.File.OpenText(camino_fileread)
        'Se asigna a la variable "linea" una linea del fichero que se abre
        'para ir localalizando el texto deseado
        linea = fichero.ReadLine

        'Mientra no se llegue al final del fichero
        While Not linea Is Nothing And bandera_InStr = False

            'Convierte la linea a mayusculas
            linea = linea.ToUpper
            'elimina los espacios del inicio y el fin de la linea, asi como el TAB
            linea = linea.Trim(" ", Chr(9))
            'sustituye la , de millares por nada
            linea = linea.Replace(",", "")
            'sustituye el . decimal por ,
            linea = linea.Replace(".", ",")
            'Continua el proceso si la linea no es un espacio
            If Not linea = "" Then
                If InStr(linea, palabra) Then

                    'bandera de aviso de que ha encontrado ya el texto deseado
                    bandera_InStr = True

                    'Una vez determinada la linea donde está "palabra" se avanzan
                    'tantas líneas como indique "inicio_lectura"

                    For i = 0 To inicio_lectura - 1
                        linea = fichero.ReadLine
                    Next
                    total_filas = 0

                    'Mientras las lineas incluyan el caracter separador de columnas...
                    While linea.Contains(car_separador_columna)
                        'La linea actual será la primera fila de la tabla
                        'sus elementos se guardan en array_fila
                        array_fila = linea.Split(car_separador_columna)


'Se eliminan espacios de inicio y fin que puedan tener los elementos de array_fila
                        For i = 0 To array_fila.Length - 1
                            array_fila(i) = array_fila(i).Trim
                        Next

 'Almacena en array_columna el elemento que ocupe la posicion "num_columna"
                        ReDim Preserve array_columna(total_filas)
                        array_columna(total_filas) = array_fila(num_columna)

                        'Lee la siguiente linea
                        linea = fichero.ReadLine
                        total_filas = total_filas + 1
                    End While


                    columna = array_columna
                End If
            End If

            'lee la siguiente linea
            linea = fichero.ReadLine

        End While

        'cierra el fichero una vez leido entero
        fichero.Close()
        'valores(0) = False
        Return columna
    End Function

    Private Function busqueda_valor_flow(ByVal camino_fileread As String, ByVal palabra As String,
ByVal columna As Integer)
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
        'Localiza una linea  que contenga "palabra". La fila se divide por el carcter";"
        'Considera elemento 0 del array aquel que contiene "palabra"
        'Devueve como string el elemento de la posición "elemento(palabra)+columna"
        '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
        Dim fichero As IO.StreamReader
        Dim linea As String
        Dim bandera_InStr As Boolean = False
        Dim array_linea() As String
        Dim posicion_palabra As Integer
        Dim m As Integer
```

```
        Dim columna_devuelta As String = Nothing


        'Se asigna a la variable "fichero" el fichero que debe abrir
        fichero = IO.File.OpenText(camino_fileread)
        'Se asigna a la variable "linea" una linea del fichero que se abre
        'para ir localalizando el texto deseado
        linea = fichero.ReadLine

        'Mientra no se llegue al final del fichero
        While Not linea Is Nothing And bandera_InStr = False

            'Convierte la linea a mayusculas
            linea = linea.ToUpper
            'elimina los espacios del inicio y el fin de la linea, asi como el TAB
            linea = linea.Trim(" ", Chr(9))
            'sustituye la , de millares por nada
            linea = linea.Replace(",", "")
            'sustituye el . decimal por ,
            linea = linea.Replace(".", ",")
            'Continua el proceso si la linea no es un espacio
            If Not linea = "" Then
                If InStr(linea, palabra) Then

                    'bandera de aviso de que ha encontrado ya el texto deseado
                    bandera_InStr = True
                    'Divide la linea en un array separando los terminos entre ;
                    array_linea = linea.Split(";")

                    'Localiza la palabra buscada entre los elementos del array anterior
                    For m = 0 To array_linea.Count - 1
                        If InStr(array_linea(m), palabra) Then
                            posicion_palabra = m
                        End If
                    Next

                    columna_devuelta = array_linea(posicion_palabra + columna)
                    columna_devuelta = columna_devuelta.Trim
                End If
            End If
            'lee la siguiente linea
            linea = fichero.ReadLine

        End While

        'cierra el fichero una vez leido entero
        fichero.Close()
        Return columna_devuelta

    End Function

    Private Function tratar_columna_flow(ByVal cadena As String, ByVal car_separador As Char, ByVal
devolver_Entero As Boolean)
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
        'Separa "cadena" por el caracter "car_separador".
        'Elimina en el segundo elemento todo lo que haya trás el primer caracter " "
        'Si se especifica TRUE devuelve los elementos del array como enteros
        'Si es FALSE se devuelven los elementos del array como string
        '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
        Dim cadena_array() As String
        Dim long_array As Integer
        Dim valores_integer(1) As Integer
        Dim valores_string() As String

'Genera un array con los elementos separados por el caracter indicado como parámtero
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
        cadena_array = cadena.Split(car_separador)
        long_array = cadena_array.Length
        ReDim Preserve valores_integer(long_array - 1)
        For h = 0 To long_array - 1
            cadena_array(h) = cadena_array(h).Trim
        Next


     'Si se va a realizar la conversión a entero
     'Eliminación de todo lo que hay tras el primer " " en el segundo elemento del array
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
        If devolver_Entero = True Then
            For h = 0 To long_array - 1
                'Try Catch sirve para controlar una excepción
                'En este caso, si la conversión a entero no es posible
                'daria excepción, de este modo se le dice que si pasa eso
                'se olvide y asigne valor 0
                '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                Try
                    valores_integer(h) = CInt(cadena_array(h))
```

```
                Catch
                    valores_integer(h) = 0
                End Try

            Next
            Return valores_integer
        Else
            valores_string = cadena_array
            Return valores_string
        End If
    End Function

    Private Function busqueda_valor_xilinx(ByVal camino_fileread As String, ByVal palabra As String)
        '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
        'Devuelve el valor numérico que hay tras la frase buscada, solo una palabra
        'Devuelve tipo single.
        'Incluye un tratamiento especial para la lectura de CPU TIME de xilinx
        '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
        Dim fichero As IO.StreamReader
        Dim linea As String
        Dim array_linea() As String
        Dim bandera_InStr As Boolean = False
        Dim valores(1) As Single
        Dim i_min As Integer
        Dim i_sec As Integer


        'Se asigna a la variable "fichero" el fichero que debe abrir
        fichero = IO.File.OpenText(camino_fileread)
        'Se asigna a la variable "linea" una linea del fichero que se abre
        'para ir localalizando el texto deseado
        linea = fichero.ReadLine
        'Mientra no se llegue al final del fichero
        While Not linea Is Nothing And bandera_InStr = False

            'Convierte la linea a mayusculas
            linea = linea.ToUpper
            'elimina los espacios del inicio y el fin de la linea, asi como el TAB
            linea = linea.Trim(" ", Chr(9))
            'sustituye la , de millares por nada
            linea = linea.Replace(",", "")
            'sustituye el . decimal por ,
            linea = linea.Replace(".", ",")
            'Continua el proceso si la linea no es un espacio
            If Not linea = "" Then
                If InStr(linea, palabra) Then

                    'bandera de aviso de que ha encontrado ya el texto deseado
                    bandera_InStr = True

'Se corta la linea prescindiendo del trozo que se le ha pasado con el parámetro "palabra"
'De este modo, queda una linea con muchos espacios por delante
'con trim se eliminan todos esos esapcios quedando solo la informacion necesaria
                    linea = linea.Substring(palabra.Length)
                    linea = linea.Trim
                    'La sublinea que queda se divide en elementos segun espacios
                    array_linea = linea.Split(" ")


'----------------------------------------------------------------------
'Se añade una condicion para el caso de estar buscando el tiempo de CPU
                    'ya que este puede ser:
                    ' 1. solo segundos (10 secs)
                    ' 2. min y seg (1 min 10 secs)
                    ' 3. solo minutos (3 min)
                    If InStr(palabra, "CPU TIME") Then
                        'Búsqueda de minutos
                        For i = 0 To array_linea.Length - 1
                            If InStr(array_linea(i), "MIN") Then
                                i_min = i
                                valores(0) = array_linea(i_min - 1)
                            Else
                                valores(0) = 0
                            End If
                        Next
                        'Búsqueda de segundos
                        For i = 0 To array_linea.Length - 1
                            'CASO 1
                            If InStr(array_linea(i), "SEC") Then
                                i_sec = i
                                valores(1) = array_linea(i_sec - 1)
                            Else
                                valores(1) = 0
                            End If
                        Next
  '---------------------------------------------------------------------------------
```

```
  'continúa aqui si no se trata de búsqueda de tiempo
  '----------------------------------------------------------------------------------
  'En ese array se conoce de antemano la posicion de los valores que se necesitan:usados
'  y disponibles
                    Else

                        '-----------------------------------------------
                        valores(0) = array_linea(0)

                        If array_linea.Length > 2 Then
                            valores(1) = array_linea(3)
                        Else
                            valores(1) = 0
                        End If

                    End If
                End If
            End If

            'lee la siguiente linea
            linea = fichero.ReadLine

        End While

        'cierra el fichero una vez leido entero
        fichero.Close()
        Return valores
    End Function

    Private Function busqueda_ultimavez(ByVal camino_fileread As String, ByVal palabra As String)
  '-------------------------------------------------------------------------------
  'No incluye bandera como en "busqueda_valor", porque se pretende que continúe buscando
  '"palabra" aunque ya la haya encontrado una vez
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´

        Dim fichero As IO.StreamReader
        Dim linea As String
        Dim array_linea() As String
        Dim bandera_InStr As Boolean = False
        Dim valores(1) As Single
        Dim i_min As Integer
        Dim i_sec As Integer


        'Se asigna a la variable "fichero" el fichero que debe abrir
        fichero = IO.File.OpenText(camino_fileread)
        'Se asigna a la variable "linea" una linea del fichero que se abre
        'para ir localizando el texto deseado
        linea = fichero.ReadLine
        'Mientra no se llegue al final del fichero
        While Not linea Is Nothing

            'Convierte la linea a mayusculas
            linea = linea.ToUpper
            'elimina los espacios del inicio y el fin de la linea, asi como el TAB
            linea = linea.Trim(" ", Chr(9))
            'sustituye la , de millares por nada
            linea = linea.Replace(",", "")
            'sustituye el . decimal por ,
            linea = linea.Replace(".", ",")
            'Continua el proceso si la linea no es un espacio
            If Not linea = "" Then
                If InStr(linea, palabra) Then

'Se corta la linea prescindiendo del trozo que se le ha pasado con el parámetro "palabra"
'De este modo, queda una linea con muchos espacios por delante
'con trim se eliminan todos esos esapcios quedando solo la informacion necesaria
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                    linea = linea.Substring(palabra.Length)
                    linea = linea.Trim

                    'La sublinea que queda se divide en elementos segun espacios
                    array_linea = linea.Split(" ")

             '----------------------------------------------------------------------
            'Se añade una condicion para el caso de estar buscando el tiempo de CPU
                    'ya que este puede ser:
                    ' 1. solo segundos (10 secs)
                    ' 2. min y seg (1 min 10 secs)
                    ' 3. solo minutos (3 min)
                    '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                    If InStr(palabra, "CPU TIME") Then
        'Búsqueda de minutos. Localiza si el array_linea tiene un elemento
        'con la palabra "MIN", si es asi devuelve el elemento anterior en el array
        'como el total de minutos mediante la variable de retorno valores(0). Si no hay
                    '"MIN" devuelve 0
```

```
                    '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                        For i = 0 To array_linea.Length - 1
                            If InStr(array_linea(i), "MIN") Then
                                i_min = i
                                valores(0) = array_linea(i_min - 1)
                            Else
                                valores(0) = 0
                            End If
                        Next

                        'Búsqueda de segundos. Similar a la búsqueda de minutos
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
                        For i = 0 To array_linea.Length - 1
                            'CASO 1
                            If InStr(array_linea(i), "SEC") Then
                                i_sec = i
                                valores(1) = array_linea(i_sec - 1)
                            Else
                                valores(1) = 0
                            End If
                        Next
'---------------------------------------------------------------------------------
'continúa aqui si no se trata de búsqueda de tiempo
'----------------------------------------------------------------------------------
      'En ese array se conoce de antemano la posicion de los valores que se necesitan:
      'usados y disponibles
                    Else

                        '-----------------------------------------------
                        valores(0) = array_linea(0)

                        If array_linea.Length > 2 Then
                            valores(1) = array_linea(3)
                        Else
                            valores(1) = 0
                        End If

                    End If
                End If
            End If

            'lee la siguiente linea
            linea = fichero.ReadLine

        End While

        'cierra el fichero una vez leido entero
        fichero.Close()
        Return valores
    End Function

    Private Function busqueda_valor_separa(ByVal camino_fileread As String, ByVal palabra As
String,ByVal caracter_separador As Char, ByVal posicion_array As Integer)
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'Localiza la linea del fichero que contiene un determinado texto eliminándolo de la misma después. A
continuación, genera un array string guardando en cada posición los elementos que se encuentren
separados por el caracter indicado mediante "caracter_separador. Por último, devuelve el elemento
situado en la posicion del array indicado por "posicion_array"
'---------------------------------------------------------------------------------------
        Dim fichero As IO.StreamReader
        Dim linea As String
        Dim array_linea() As String
        Dim bandera_InStr As Boolean = False
        Dim valor As Single

        'Se asigna a la variable "fichero" el fichero que debe abrir
        fichero = IO.File.OpenText(camino_fileread)
        'Se asigna a la variable "linea" una linea del fichero que se abre
        'para ir localalizando el texto deseado
        linea = fichero.ReadLine

        'Mientra no se llegue al final del fichero
        While Not linea Is Nothing And bandera_InStr = False

            'Convierte la linea a mayusculas
            linea = linea.ToUpper
            'elimina los espacios del inicio y el fin de la linea, asi como el TAB
            linea = linea.Trim(" ", Chr(9))
            'sustituye la , de millares por nada
            linea = linea.Replace(",", "")
            'sustituye el . decimal por ,
            linea = linea.Replace(".", ",")
            'Continua el proceso si la linea no es un espacio
            If Not linea = "" Then
                If InStr(linea, palabra) Then
```

```
                    'bandera de aviso de que ha encontrado ya el texto deseado
                    bandera_InStr = True

'Se corta la linea prescindiendo del trozo que se le ha pasado con el parámetro "palabra". De este
modo, queda una linea con muchos espacios por delante                'con trim se eliminan todos esos
esapcios quedando solo la informacion necesaria
                    linea = linea.Substring(palabra.Length)
                    linea = linea.Trim
                    'La sublinea que queda se divide en elementos segun espacios
                    array_linea = linea.Split(caracter_separador)
                    For i = 0 To array_linea.Length - 1
                        array_linea(i) = array_linea(i).Trim
                    Next

 'Si existe el fichero pero no hay ningun valor de potencia calculado
 'detecta el array sin ningun valor lo que da errror
 Para evitarlo se evalúa el contenido de la posicion especificada del array antes de
                    'asignar el valor a la variable de retorno de la funcion
                    If posicion_array > (array_linea.Length - 1) Then
                        valor = 0
                    ElseIf array_linea(posicion_array) = "" Then
                        valor = 0
                    Else
                        valor = array_linea(posicion_array)
                    End If
                End If
            End If

            'lee la siguiente linea
            linea = fichero.ReadLine

        End While

        'cierra el fichero una vez leido entero
        fichero.Close()
        Return valor
    End Function



    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
Me.Bench_consumoTableAdapter.Fill(Me.Resultados_benchamrkDataSet.bench_consumo)
Me.Bench_comunicacionesTableAdapter.Fill(Me.Resultados_benchamrkDataSet.bench_comunicaciones
Me.Bench_autoTableAdapter.Fill(Me.Resultados_benchamrkDataSet.bench_auto)
Me.Circuitos_implementadosTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Circuitos_implementados)
Me.Circuitos_implementadosTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Circuitos_implementados)
Me.DatosReportsTableAdapter.Fill(Resultados_benchamrkDataSet.DatosReports)
Me.SobredimensionadosTableAdapter.Fill(Resultados_benchamrkDataSet.Sobredimensionados)
Me.FallosTableAdapter.Fill(Resultados_benchamrkDataSet.Fallos)

        TextBox2.Text = DatosReportsBindingSource.Count
        TextBox3.Text = FallosBindingSource.Count
        TextBox4.Text = SobredimensionadosBindingSource.Count
End Sub

Private Sub ToolStripButton2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ToolStripButton2.Click
        Application.Exit()
End Sub

    Private Sub LecturaDeInformesToolStripMenuItem_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles LecturaDeInformesToolStripMenuItem.Click
Me.DatosReportsTableAdapter.Fill(Resultados_benchamrkDataSet.DatosReports)
Me.SobredimensionadosTableAdapter.Fill(Resultados_benchamrkDataSet.Sobredimensionados)
Me.FallosTableAdapter.Fill(Resultados_benchamrkDataSet.Fallos)

        TextBox2.Text = DatosReportsBindingSource.Count
        TextBox3.Text = FallosBindingSource.Count
        TextBox4.Text = SobredimensionadosBindingSource.Count
    End Sub

    Private Sub PuntuacionesToolStripMenuItem_Click(ByVal sender As System.Object,_
    ByVal e As System.EventArgs) Handles PuntuacionesToolStripMenuItem.Click
        Form2.Show()
        Me.Hide()
    End Sub

    Private Sub CircuitosImplementadosToolStripMenuItem_Click(ByVal sender As System.Object,ByVal e
As System.EventArgs) Handles CircuitosImplementadosToolStripMenuItem.Click
        Form3.Show()
        Me.Hide()
    End Sub

    Private Sub CircuitosPorSectorToolStripMenuItem_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles CircuitosPorSectorToolStripMenuItem.Click
```

```
        Form5.Show()
        Me.Hide()
    End Sub
End Class
```

FORM2.VB

```
Public Class Form2
    Public Structure Puntuaciones
        Public puntuaciones_id As Integer
        Public puntuaciones_circuito As String
        Public puntuaciones_estrategia As String
        Public puntuaciones_tecnologia As String
        Public puntuaciones_fmax As Single
        Public puntuaciones_ul_sobran As Single
        Public puntuaciones_bits_sobran As Single
        Public puntuaciones_mult_sobran As Single
        Public puntuaciones_pot As Single
        Public puntuaciones_precio As Single
        Public puntuaciones_tcpu As Single
        Public puntuaciones_memo As Single
        Public puntuaciones_P1 As Single
        Public puntuaciones_P2 As Single
        Public puntuaciones_P3 As Single
        Public puntuaciones_P4 As Single
        Public puntuaciones_P5 As Single
    End Structure

    Public Structure Normalizados
        Public normalizados_circuito As String
        Public normalizados_estrategia As String
        Public normalizados_tecnologia As String
        Public normalizados_P1_N As Single
        Public normalizados_P2_N As Single
        Public normalizados_P3_N As Single
        Public normalizados_P4_N As Single
        Public normalizados_P5_N As Single
    End Structure

    Public Structure fallos_puntuaciones
        Public fallo_puntuaciones_circuito As String
        Public fallo_puntuaciones_estrategia As String
        Public fallo_puntuaciones_tecnologia As String
        Public fallo_puntuaciones_precio As Boolean
        Public fallo_puntuaciones_area As Boolean
        Public fallo_puntuaciones_fmax As Boolean
        Public fallo_puntuaciones_pot As Boolean
        Public fallo_puntuaciones_Tcpu As Boolean
        Public fallo_puntuaciones_memo As Boolean
    End Structure

    Public Structure puntuaciones_min_max
        Public puntuaciones_min_max_circuito As String
        Public puntuaciones_min_max_estrategia As String
        Public puntuaciones_min_max_P1_min As Single
        Public puntuaciones_min_max_P1_max As Single
        Public puntuaciones_min_max_P2_min As Single
        Public puntuaciones_min_max_P2_max As Single
        Public puntuaciones_min_max_P3_min As Single
        Public puntuaciones_min_max_P3_max As Single
        Public puntuaciones_min_max_P4_min As Single
        Public puntuaciones_min_max_P4_max As Single
        Public puntuaciones_min_max_P5_min As Single
        Public puntuaciones_min_max_P5_max As Single
    End Structure


    Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Me.PuntuacionesTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Puntuaciones)
        Me.NormalizadosTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Normalizados)
Me.Fallos_puntuacionTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Fallos_puntuacion)

        TextBox1.Text = PuntuacionesBindingSource.Count
        TextBox2.Text = NormalizadosBindingSource.Count
        TextBox3.Text = Fallos_puntuacionBindingSource.Count
    End Sub

    Private Sub ToolStripButton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ToolStripButton1.Click
```

```
        'Mediante FILL se rellenan las tablas con las que necesitamos trabajar: DatosReports y
Caracteristicas
        Me.DatosReportsTableAdapter.Fill(Me.Resultados_benchamrkDataSet.DatosReports)
Me.CaracteristicasTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Caracteristicas)
        Me.PuntuacionesTableAdapter.DeleteQuery()
        Me.Fallos_puntuacionTableAdapter.DeleteQuery()
        Me.CircuitoTableAdapter.DeleteQuery()
        Me.Puntuaciones_min_maxTableAdapter.DeleteQuery()
        Me.NormalizadosTableAdapter.DeleteQuery()


        '''''''''''''''''''''''''''''''''''''''''''''
        ' TABLAS PUNTUACIONES Y FALLOS_PUNTUACIONES '
        '''''''''''''''''''''''''''''''''''''''''''''
        Dim registro_puntuaciones As Puntuaciones
        Dim registro_fallos_puntuaciones As fallos_puntuaciones
        Dim total_datosreports As Integer
        Dim calculo_UL_sobran As Single
        Dim calculo_bits_sobran As Single
        Dim calculo_mult_sobran As Single
        Dim fmax As Single
        Dim puntuacion_2 As Single
        Dim puntuacion_3 As Single
        Dim pot As Single
        Dim tcpu As Single
        Dim tsint As Single
        Dim timp As Single
        Dim fila_datosreports As DataRowView
        Dim memo As Single
        Dim tecnologia As String
        Dim tecnologia_ind As String = 0
        Dim tecnologia_com As String = 0

        'Bindingsource es el puntero que recorre la TablaAdapter
        'Lleva tambien la cuenta total de registros
        total_datosreports = DatosReportsBindingSource.Count

        'Se realiza un bucle para ir recorriendo todos los registros de DatosReports
        'y extraer de cada uno el campo que interese
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        For i = 0 To total_datosreports - 1
            'Lee el registro indicado,i, y lo almacena en el datarrowview
            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
            fila_datosreports = DatosReportsBindingSource.Item(i)

            'El datarowview tiene un componente que es ItemArray, donde realmente
            'está guardados los valores del registro
            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
            'Los siguientes campos se copian sin mas desde DatosReports a Puntuaciones
            registro_puntuaciones.puntuaciones_circuito = fila_datosreports.Row.ItemArray(2)
            registro_puntuaciones.puntuaciones_estrategia = fila_datosreports.Row.ItemArray(3)

            'Se añadirán dos registros para xilinx cuya única diferencia será el sufijo
            'relativo a la temperatura. Se comprueba si son estandar o de automocion
            'ya que usan letras diferentes

            ''''''''''''''''''
            'CAMPO TECNOLOGIA'
            ''''''''''''''''''
            tecnologia = fila_datosreports.Row.ItemArray(4)
            tecnologia = tecnologia.ToUpper
            If tecnologia.ElementAt(0) = "X" Then
                If tecnologia.ElementAt(1) = "A" Then
                    tecnologia_ind = tecnologia + "Q"
                    tecnologia_com = tecnologia + "I"
                ElseIf tecnologia.ElementAt(1) = "C" Then
                    tecnologia_ind = tecnologia + "I"
                    tecnologia_com = tecnologia + "C"
                End If
            End If

            ''''''''''''''''''
            'CAMPO UL sobran '
            ''''''''''''''''''
            '% de Unidades lógicas que sobran
            If fila_datosreports.Row.ItemArray(7) <> 0 Then
                calculo_UL_sobran = 1 - (fila_datosreports.Row.ItemArray(6) / _
                fila_datosreports.Row.ItemArray(7))
                calculo_UL_sobran = calculo_UL_sobran * 100
            Else
                calculo_UL_sobran = 100
            End If


            ''''''''''''''''''
            'CAMPO bits sobran '
```

```
            ''''''''''''''''''
            '% de bits de memoria utilizados
            If fila_datosreports.Row.ItemArray(9) <> 0 Then
                calculo_bits_sobran = 1 - (fila_datosreports.Row.ItemArray(8) / _
                fila_datosreports.Row.ItemArray(9))
                calculo_bits_sobran = calculo_bits_sobran * 100
            Else
                'si no se ha usado memoria se conserva el 100% de los bits iniciales
                calculo_bits_sobran = 100
            End If


            '''''''''''''''''''''''''
            'CAMPO mult18x18 sobran '
            '''''''''''''''''''''''''
            '% de multiplicadores utilizados
            If fila_datosreports.Row.ItemArray(11) <> 0 Then
                calculo_mult_sobran = 1 - (fila_datosreports.Row.ItemArray(10) / _
                fila_datosreports.Row.ItemArray(11))
                calculo_mult_sobran = calculo_mult_sobran * 100
            Else
                calculo_mult_sobran = 100
            End If


            'Si algunos de los recursos usados sobrepasa a los disponibles total se almacenará en
latabla de fallos de puntuaciones
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
            If (calculo_UL_sobran < 0 Or calculo_mult_sobran < 0 Or calculo_bits_sobran < 0) Then
                registro_fallos_puntuaciones.fallo_puntuaciones_area = True
            Else
                registro_puntuaciones.puntuaciones_ul_sobran = calculo_UL_sobran
                registro_puntuaciones.puntuaciones_bits_sobran = calculo_bits_sobran
                registro_puntuaciones.puntuaciones_mult_sobran = calculo_mult_sobran
            End If

            ''''''''''''''
            ' CAMPO FMAX '
            ''''''''''''''
            If fila_datosreports.Row.ItemArray(15) <> 0 Then
                fmax = (1 / fila_datosreports.Row.ItemArray(15)) * 1000
                registro_puntuaciones.puntuaciones_fmax = fmax
            Else
                registro_fallos_puntuaciones.fallo_puntuaciones_fmax = True
            End If

            ''''''''''''''
            ' CAMPO POT  '
            ''''''''''''''
            pot = fila_datosreports.Row.ItemArray(14)
            If pot <> 0 Then
                registro_puntuaciones.puntuaciones_pot = pot
            Else
                registro_fallos_puntuaciones.fallo_puntuaciones_pot = True
            End If

            '''''''''''''''''
            ' CAMPO PRECIO  '
            '''''''''''''''''
            'Buscar en la tabla CARACTERISTICAS las dos tecnologias determinadas para la tecnología
especificada en DatosReport. Cuando coincida devuelve el campo PRECIO
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            Dim fila_caracteristicas As DataRowView
            Dim total_caracteristicas As Integer
            Dim precio_ind As Single
            Dim precio_com As Single
            Dim precio_altera As Single

            'Número total de registros de la tabla CARACTERISTICAS
            total_caracteristicas = CaracteristicasBindingSource.Count
            fila_caracteristicas = CaracteristicasBindingSource.Item(0)

            If tecnologia.StartsWith("X") Then
precio_ind = busqueda_precio(tecnologia_ind, fila_caracteristicas,total_caracteristicas)
                If precio_ind = 0 Then
        registro_fallos_puntuaciones.fallo_puntuaciones_precio = True
        registro_fallos_puntuaciones.fallo_puntuaciones_tecnologia = tecnologia_ind
                End If
precio_com = busqueda_precio(tecnologia_com, fila_caracteristicas,total_caracteristicas)
                If precio_com = 0 Then
            registro_fallos_puntuaciones.fallo_puntuaciones_precio = True
            registro_fallos_puntuaciones.fallo_puntuaciones_tecnologia = tecnologia_com
                End If

            Else
precio_altera = busqueda_precio(tecnologia, fila_caracteristicas, total_caracteristicas)
```

```
        If precio_altera = 0 Then
         registro_fallos_puntuaciones.fallo_puntuaciones_precio = True
         registro_fallos_puntuaciones.fallo_puntuaciones_tecnologia = tecnologia
        End If
    End If


    '''''''''''''''''''''''
    ' CAMPO P1=F/precio   '
    '''''''''''''''''''''''
    Dim P1_ind As Single
    Dim P1_com As Single
    Dim P1_altera As Single

    If tecnologia.StartsWith("X") Then
        P1_ind = fmax / precio_ind
        P1_com = fmax / precio_com
    Else
        P1_altera = fmax / precio_altera
    End If

    'Comprueba si el resultado es infinita, en ese caso susituye el valor por "-1"
    If (Single.IsInfinity(puntuacion_2)) Then puntuacion_2 = -1
    End If

    '''''''''''''''''''''''''''''''''
    ' CAMPO P2=%UL_sobran/precio '
    '''''''''''''''''''''''''''''''''
    Dim P2_ind As Single
    Dim P2_com As Single
    Dim P2_altera As Single

    If tecnologia.StartsWith("X") Then
        P2_ind = calculo_UL_sobran / precio_ind
        P2_com = calculo_UL_sobran / precio_com
    Else
        P2_altera = calculo_UL_sobran / precio_altera
    End If

    '''''''''''''''''''''''''''''''''
    ' CAMPO P3=%bits_sobran/precio '
    '''''''''''''''''''''''''''''''''
    Dim P3_ind As Single
    Dim P3_com As Single
    Dim P3_altera As Single
    If tecnologia.StartsWith("X") Then
        P3_ind = calculo_bits_sobran / precio_ind
        P3_com = calculo_bits_sobran / precio_com
    Else
        P3_altera = calculo_bits_sobran / precio_altera
    End If

    '''''''''''''''''''''''''''''''''
    ' CAMPO P4=%mult_sobran/precio '
    '''''''''''''''''''''''''''''''''
    Dim P4_ind As Single
    Dim P4_com As Single
    Dim P4_altera As Single
    If tecnologia.StartsWith("X") Then
        P4_ind = calculo_mult_sobran / precio_ind
        P4_com = calculo_mult_sobran / precio_com
    Else
        P4_altera = calculo_bits_sobran / precio_altera
    End If

    '''''''''''''''''''''''''''''''''
    ' CAMPO P5=( 1/potencia)/precio     '
    '''''''''''''''''''''''''''''''''
    'Se convierte la potencia de mW a W dividiendo por 1000 el valor almacenado

    Dim P5_ind As Single
    Dim P5_com As Single
    Dim P5_altera As Single

    If tecnologia.StartsWith("X") Then
        P5_ind = (1 / (pot / 1000)) / precio_ind
        P5_com = (1 / (pot / 1000)) / precio_com
    Else
        P5_altera = (1 / (pot / 1000)) / precio_altera
    End If

    '''''''''''''''
    ' CAMPO TCPU '
    '''''''''''''''
    tsint = fila_datosreports.Row.ItemArray(19)
    timp = fila_datosreports.Row.ItemArray(20)
```

```
              tcpu = tsint + timp

              If tcpu <> 0 Then
                  registro_puntuaciones.puntuaciones_tcpu = tcpu
              Else
                  registro_puntuaciones.puntuaciones_tcpu = True
              End If

              ''''''''''''''
              ' CAMPO MEMO '
              ''''''''''''''
              memo = fila_datosreports.Row.ItemArray(21)
              If memo <> 0 Then
                  registro_puntuaciones.puntuaciones_memo = memo
              Else
                  registro_fallos_puntuaciones.fallo_puntuaciones_memo = 0
              End If

              ''''''''''''''
              ' CAMPO P3   '
              ''''''''''''''
              puntuacion_3 = tcpu * memo
              registro_puntuaciones.puntuaciones_P3 = puntuacion_3
            ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'Completada las estructura se inserta el registro en la tabla Puntuaciones, para Xilinx
'se insertan dos registros, una para el modelo comercial y otro para el industrial
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'Si alguno de los parámetros resultó un valor 0, el registro se guardará en la tabla            'fallo
puntuaciones, evitándose asi resultados INFINITO en el cálculo de puntuaciones
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
registro_fallos_puntuaciones.fallo_puntuaciones_circuito=fila_datosreports.Row.ItemArray(2)
registro_fallos_puntuaciones.fallo_puntuaciones_estrategia=fila_datosreports.Row.ItemArray(3)

              'si la tecnologia corresponde a xilinx (X)
              If tecnologia.ElementAt(0) = "X" Then
                  ''''''''''''''''''''
                  ' XILINX
                  ''''''''''''''''''''''
                  'Industrial
                  ''''''''''''
                  registro_puntuaciones.puntuaciones_tecnologia = tecnologia_ind
                  If precio_ind = 0 Then
               registro_fallos_puntuaciones.fallo_puntuaciones_precio = True
               registro_fallos_puntuaciones.fallo_puntuaciones_tecnologia = tecnologia_ind
                  ElseIf registro_fallos_puntuaciones.fallo_puntuaciones_fmax = True Then
               registro_fallos_puntuaciones.fallo_puntuaciones_tecnologia = tecnologia_ind
                  ElseIf registro_fallos_puntuaciones.fallo_puntuaciones_pot = True Then
               registro_fallos_puntuaciones.fallo_puntuaciones_tecnologia = tecnologia_ind
                  Else
                      registro_puntuaciones.puntuaciones_precio = precio_ind
                      registro_puntuaciones.puntuaciones_P1 = P1_ind
                      registro_puntuaciones.puntuaciones_P2 = P2_ind
                      registro_puntuaciones.puntuaciones_P3 = P3_ind
                      registro_puntuaciones.puntuaciones_P4 = P4_ind
                      registro_puntuaciones.puntuaciones_P5 = P5_ind
                  End If


  'Si el precio_ind no es 0 ni nigun otro valor, añade el modelo de xilinx INDUSTRIAL a PUNTUACIONES en
  caso contrario lo añada a FALLO_PUNTUACIONES
  '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´,,,,,,,,,,,,,,,,,,,,,,
                  If (registro_fallos_puntuaciones.fallo_puntuaciones_precio Or _
                      registro_fallos_puntuaciones.fallo_puntuaciones_area Or _
                      registro_fallos_puntuaciones.fallo_puntuaciones_fmax Or _
                      registro_fallos_puntuaciones.fallo_puntuaciones_pot Or _
                      registro_fallos_puntuaciones.fallo_puntuaciones_Tcpu Or _
                      registro_fallos_puntuaciones.fallo_puntuaciones_memo) Then

  Fallos_puntuacionTableAdapter.Insert(registro_fallos_puntuaciones.fallo_puntuaciones_circuito,
  registro_fallos_puntuaciones.fallo_puntuaciones_estrategia, _
  registro_fallos_puntuaciones.fallo_puntuaciones_tecnologia, _
           registro_fallos_puntuaciones.fallo_puntuaciones_precio, _
           registro_fallos_puntuaciones.fallo_puntuaciones_area, _
           registro_fallos_puntuaciones.fallo_puntuaciones_fmax, _
           registro_fallos_puntuaciones.fallo_puntuaciones_pot, _
           registro_fallos_puntuaciones.fallo_puntuaciones_Tcpu, _
           registro_fallos_puntuaciones.fallo_puntuaciones_memo)
                  Else
  PuntuacionesTableAdapter.Insert(registro_puntuaciones.puntuaciones_circuito, _
  registro_puntuaciones.puntuaciones_estrategia, _
                      registro_puntuaciones.puntuaciones_tecnologia, _
                      registro_puntuaciones.puntuaciones_fmax, _
                      registro_puntuaciones.puntuaciones_ul_sobran, _
                      registro_puntuaciones.puntuaciones_bits_sobran, _
                      registro_puntuaciones.puntuaciones_mult_sobran, _
                      registro_puntuaciones.puntuaciones_pot, _
```

```
                    registro_puntuaciones.puntuaciones_precio, _
                    registro_puntuaciones.puntuaciones_tcpu, _
                    registro_puntuaciones.puntuaciones_memo, _
registro_puntuaciones.puntuaciones_P1, _
registro_puntuaciones.puntuaciones_P2, _
registro_puntuaciones.puntuaciones_P3, _
registro_puntuaciones.puntuaciones_P4, _
registro_puntuaciones.puntuaciones_P5)
                End If
 'Se resetea esta variable para el siguiente elemento a insertar en esta misma iteracion
 '(Xilinx añade dos precios uno para C y otro para I)
                registro_fallos_puntuaciones.fallo_puntuaciones_precio = False

                ''''''''''''''''''''''
                'Comercial
                ''''''''''''
                registro_puntuaciones.puntuaciones_tecnologia = tecnologia_com
                If precio_com = 0 Then
            registro_fallos_puntuaciones.fallo_puntuaciones_precio = True
            registro_fallos_puntuaciones.fallo_puntuaciones_tecnologia = tecnologia_com
                ElseIf registro_fallos_puntuaciones.fallo_puntuaciones_fmax = True Then
            registro_fallos_puntuaciones.fallo_puntuaciones_tecnologia = tecnologia_com
                ElseIf registro_fallos_puntuaciones.fallo_puntuaciones_pot = True Then
             registro_fallos_puntuaciones.fallo_puntuaciones_tecnologia = tecnologia_com
                Else
                    registro_puntuaciones.puntuaciones_precio = precio_com
                    registro_puntuaciones.puntuaciones_P1 = P1_com
                    registro_puntuaciones.puntuaciones_P2 = P2_com
                    registro_puntuaciones.puntuaciones_P3 = P3_com
                    registro_puntuaciones.puntuaciones_P4 = P4_com
                    registro_puntuaciones.puntuaciones_P5 = P5_com
                End If

                'Si el precio_ind no es 0 ni nigun otro valor, añade el modelo de xilinx COMERCIAL a
PUNTUACIONES en caso contrario lo añada a FALLO_PUNTUACIONES
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
                If (registro_fallos_puntuaciones.fallo_puntuaciones_precio Or _
                    registro_fallos_puntuaciones.fallo_puntuaciones_area Or _
                    registro_fallos_puntuaciones.fallo_puntuaciones_fmax Or _
                    registro_fallos_puntuaciones.fallo_puntuaciones_pot Or _
                    registro_fallos_puntuaciones.fallo_puntuaciones_Tcpu Or _
                    registro_fallos_puntuaciones.fallo_puntuaciones_memo) Then

Fallos_puntuacionTableAdapter.Insert(registro_fallos_puntuaciones.fallo_puntuaciones_circuito,registr
o_fallos_puntuaciones.fallo_puntuaciones_precio, _
                 registro_fallos_puntuaciones.fallo_puntuaciones_area, _
                 registro_fallos_puntuaciones.fallo_puntuaciones_fmax, _
                 registro_fallos_puntuaciones.fallo_puntuaciones_pot, _
                 registro_fallos_puntuaciones.fallo_puntuaciones_Tcpu, _
                 registro_fallos_puntuaciones.fallo_puntuaciones_memo)
                Else
PuntuacionesTableAdapter.Insert(registro_puntuaciones.puntuaciones_circuito, _
registro_puntuaciones.puntuaciones_estrategia, _
                    registro_puntuaciones.puntuaciones_tecnologia, _
                    registro_puntuaciones.puntuaciones_fmax, _
                    registro_puntuaciones.puntuaciones_ul_sobran, _
                    registro_puntuaciones.puntuaciones_bits_sobran, _
                    registro_puntuaciones.puntuaciones_mult_sobran, _
                    registro_puntuaciones.puntuaciones_pot, _
                    registro_puntuaciones.puntuaciones_precio, _
                    registro_puntuaciones.puntuaciones_tcpu, _
                    registro_puntuaciones.puntuaciones_memo, _
registro_puntuaciones.puntuaciones_P1, _
registro_puntuaciones.puntuaciones_P2, _
registro_puntuaciones.puntuaciones_P3, _
registro_puntuaciones.puntuaciones_P4, _
registro_puntuaciones.puntuaciones_P5)
                End If

 'Se resetea esta variable para el siguiente elemento a insertar en esta misma iteracion
 '(Xilinx añade dos precios uno para C y otro para I)
                registro_fallos_puntuaciones.fallo_puntuaciones_precio = False

            Else
                ''''''''''''''''
                '  ALTERA     '
                ''''''''''''''''
                registro_puntuaciones.puntuaciones_tecnologia = tecnologia
                registro_puntuaciones.puntuaciones_precio = precio_altera
                registro_puntuaciones.puntuaciones_P1 = P1_altera
                registro_puntuaciones.puntuaciones_P2 = P2_altera
                registro_puntuaciones.puntuaciones_P3 = P3_altera
                registro_puntuaciones.puntuaciones_P4 = P4_altera
                registro_puntuaciones.puntuaciones_P5 = P5_altera

                If (registro_fallos_puntuaciones.fallo_puntuaciones_precio Or _
```

```
                    registro_fallos_puntuaciones.fallo_puntuaciones_area Or _
                    registro_fallos_puntuaciones.fallo_puntuaciones_fmax Or _
                    registro_fallos_puntuaciones.fallo_puntuaciones_pot Or _
                    registro_fallos_puntuaciones.fallo_puntuaciones_Tcpu Or _
                    registro_fallos_puntuaciones.fallo_puntuaciones_memo)
                  Then
        Fallos_puntuacionTableAdapter.Insert(registro_fallos_puntuaciones.fallo_puntuaciones_circuit
        o,registro_fallos_puntuaciones.fallo_puntuaciones_estrategia, _
        registro_fallos_puntuaciones.fallo_puntuaciones_tecnologia, _
        registro_fallos_puntuaciones.fallo_puntuaciones_precio, _
        registro_fallos_puntuaciones.fallo_puntuaciones_area, _
        registro_fallos_puntuaciones.fallo_puntuaciones_fmax, _
        registro_fallos_puntuaciones.fallo_puntuaciones_pot, _
        registro_fallos_puntuaciones.fallo_puntuaciones_Tcpu, _
        registro_fallos_puntuaciones.fallo_puntuaciones_memo)
                Else
PuntuacionesTableAdapter.Insert(registro_puntuaciones.puntuaciones_circuito, _
registro_puntuaciones.puntuaciones_estrategia, _
registro_puntuaciones.puntuaciones_tecnologia, _
registro_puntuaciones.puntuaciones_fmax, _
registro_puntuaciones.puntuaciones_ul_sobran, _
registro_puntuaciones.puntuaciones_bits_sobran, _
registro_puntuaciones.puntuaciones_mult_sobran, _
registro_puntuaciones.puntuaciones_pot, _
registro_puntuaciones.puntuaciones_precio, _
registro_puntuaciones.puntuaciones_tcpu, _
registro_puntuaciones.puntuaciones_memo, _
registro_puntuaciones.puntuaciones_P1, _
registro_puntuaciones.puntuaciones_P2, _
registro_puntuaciones.puntuaciones_P3, _
registro_puntuaciones.puntuaciones_P4, _
registro_puntuaciones.puntuaciones_P5)
                End If
            End If

       'Se resetea el valor del registro insertado en esta iteración, por cada tabla
       '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            registro_fallos_puntuaciones.fallo_puntuaciones_circuito = ""
            registro_fallos_puntuaciones.fallo_puntuaciones_estrategia = ""
            registro_fallos_puntuaciones.fallo_puntuaciones_tecnologia = ""
            registro_fallos_puntuaciones.fallo_puntuaciones_precio = False
            registro_fallos_puntuaciones.fallo_puntuaciones_area = False
            registro_fallos_puntuaciones.fallo_puntuaciones_fmax = False
            registro_fallos_puntuaciones.fallo_puntuaciones_pot = False
            registro_fallos_puntuaciones.fallo_puntuaciones_Tcpu = False
            registro_fallos_puntuaciones.fallo_puntuaciones_memo = False

            registro_puntuaciones.puntuaciones_circuito = ""
            registro_puntuaciones.puntuaciones_estrategia = ""
            registro_puntuaciones.puntuaciones_tecnologia = ""
            registro_puntuaciones.puntuaciones_mult_sobran = 0
            registro_puntuaciones.puntuaciones_bits_sobran = 0
            registro_puntuaciones.puntuaciones_ul_sobran = 0
            registro_puntuaciones.puntuaciones_fmax = 0
            registro_puntuaciones.puntuaciones_pot = 0
            registro_puntuaciones.puntuaciones_P1 = 0
            registro_puntuaciones.puntuaciones_precio = 0
            registro_puntuaciones.puntuaciones_P2 = 0
            registro_puntuaciones.puntuaciones_tcpu = 0
            registro_puntuaciones.puntuaciones_memo = 0
            registro_puntuaciones.puntuaciones_P3 = 0
            registro_puntuaciones.puntuaciones_P4 = 0
            registro_puntuaciones.puntuaciones_P5 = 0

        Next

        Me.PuntuacionesTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Puntuaciones)
Me.Fallos_puntuacionTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Fallos_puntuacion)

        TextBox1.Text = PuntuacionesBindingSource.Count
        TextBox3.Text = Fallos_puntuacionBindingSource.Count

        ''''''''''''''''''''
        ' TABLA CIRCUITOS  '
        ''''''''''''''''''''
'Determina el número total de valor_columna existentes en el benchmark y los guarda en tabla
valor_columna
        Dim array_valor_columna() As String
        Dim registro_circuito As String
        Dim total_valor_columna As Integer
        array_valor_columna = conjunto_valor_columna(PuntuacionesBindingSource, 1)

 'Recorre el array obtenido de lafuncion para insertarlos en la tabla correspondiente
        total_valor_columna = array_valor_columna.Count
        For i = 0 To total_valor_columna - 1
            registro_circuito = array_valor_columna(i)
```

```
            CircuitoTableAdapter.Insert(registro_circuito)
        Next


        ''''''''''''''''''''''''''''''''''''''''´'''''''''''''''''''''''''''''
        '               CALCULO DE PUNTUACIONES_MIN_MAX (Y=mx+b)
        'Con max y min se determinarán m y b, pudiendo establecer luego la    '
        'relación. Y->valor de 0 a 10 y x->puntuacion original (parametro/$)  '
        'P1_N = P1 (1 a 10) ; P2_N = P2 (1 a 10)  ; P3_N = P3 (1 a 10)
        'P4_= P4 (1 a 10) ; P5_N = P5 (1 a 10)
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        Dim aux(1) As Single
        Dim total_puntuaciones As Integer
        Dim contador As Integer
        Dim matriz_c_e_P1(11, 0) As String
        Dim fila_puntuaciones As DataRowView
        Dim long_matriz As Integer
        Dim contador_2 As Integer
        Dim aviso_encontrado As Boolean
        Dim registro_puntuaciones_min_max As puntuaciones_min_max
        total_puntuaciones = PuntuacionesBindingSource.Count
        If total_puntuaciones > 0 Then
            'Carga el primer registro de puntuaciones
            fila_puntuaciones = PuntuacionesBindingSource.Item(0)
‘Guarda el contenido de los campos circuito, estrategia, p1,p2...p5 del registro 0
'Guarda P1, P2... por duplicado para ir acumulando el valor máximo y el minimo. El ‘proceso siguiente
irá analizando cada registro si los campos C y E ya estan guardados ‘en la matriz, comprobará si los
campos de puntuación contienen un valor inferior o ‘superior hasta el momento almacenados como min y
max por cada puntuación y circuito
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        '   C1      C2      C3  ...
        '   E1      E1      E2  ...
        '   P1min  P1min    P1min
        '   P1max   P1max   P1max
        '   P2min   P2min   P3min
        '   ...     ...     ...
        '   P5max   P5max   P5max
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        contador = 0
            matriz_c_e_P1(0, contador) = fila_puntuaciones.Row.ItemArray(1)
            matriz_c_e_P1(1, contador) = fila_puntuaciones.Row.ItemArray(2)
            matriz_c_e_P1(2, contador) = fila_puntuaciones.Row.ItemArray(12)
            matriz_c_e_P1(3, contador) = fila_puntuaciones.Row.ItemArray(12)
            matriz_c_e_P1(4, contador) = fila_puntuaciones.Row.ItemArray(13)
            matriz_c_e_P1(5, contador) = fila_puntuaciones.Row.ItemArray(13)
            matriz_c_e_P1(6, contador) = fila_puntuaciones.Row.ItemArray(14)
            matriz_c_e_P1(7, contador) = fila_puntuaciones.Row.ItemArray(14)
            matriz_c_e_P1(8, contador) = fila_puntuaciones.Row.ItemArray(15)
            matriz_c_e_P1(9, contador) = fila_puntuaciones.Row.ItemArray(15)
            matriz_c_e_P1(10, contador) = fila_puntuaciones.Row.ItemArray(16)
            matriz_c_e_P1(11, contador) = fila_puntuaciones.Row.ItemArray(16)

            'Carga cada registro de Puntuaciones y comprueba si sus campos circuito y
            'estrategia están ya presentes en la matriz
            For i = 1 To total_puntuaciones - 1
                aviso_encontrado = False
                fila_puntuaciones = PuntuacionesBindingSource.Item(i)

      'Recorre la matriz para comprobar si circuito y estrategias ya existen en la misma
      'la lngitud de la matriz (nº total de columnas) es long_matriz
                long_matriz = matriz_c_e_P1.GetLength(1)
                contador_2 = 0
                While (contador_2 < long_matriz And aviso_encontrado = False)
                    'Si el circuito actual de puntuaciones es igual al actual de matriz
                  If matriz_c_e_P1(0, contador_2) = fila_puntuaciones.Row.ItemArray(1) Then
               'Comprueba si las estrategias tambien coinciden
               If matriz_c_e_P1(1, contador_2) = fila_puntuaciones.Row.ItemArray(2) Then
               'Comprueba si el valor de P1 almacenado en (2) es mayor que el de fila
               ‘puntuaciones actual previamente lo convierte de string a single
                 ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                            'P1_MIN'
                            ''''''''
       'Si el nuevo valor es menor que el almacenado
        If CSng(fila_puntuaciones.Row.ItemArray(12)) < CSng(matriz_c_e_P1(2, contador_2)) Then
       'Si es menor almacena este nuevo valor P1 susituyendo al anterior existente en
       'matriz para ese valor_columna y estrategia
        matriz_c_e_P1(2, contador_2) = fila_puntuaciones.Row.ItemArray(12)
       End If
                            ''''''''
                            'P1_MAX'
                            ''''''''
       'Comprueba si P1 nuevo es mayor q el almacenado como tal
        If CSng(fila_puntuaciones.Row.ItemArray(12)) > CSng(matriz_c_e_P1(3, contador_2)) Then
        'Si es asi sustituye al anterior existente en la
       'matriz para ese valor_columna y estrategia
       matriz_c_e_P1(3, contador_2) = fila_puntuaciones.Row.ItemArray(12)
```

```
    End If
                        'P2_MIN'
                        ''''''''
      'Si el nuevo valor es menor que el almacenado
     If CSng(fila_puntuaciones.Row.ItemArray(13)) < CSng(matriz_c_e_P1(4, contador_2)) Then
    'Si es menor almacena este nuevo valor P1 susituyendo al anterior existente en
    'matriz para ese valor_columna y estrategia
    matriz_c_e_P1(4, contador_2) = fila_puntuaciones.Row.ItemArray(13)
    End If
                        ''''''''
                        'P2_MAX'
                        ''''''''
    'Comprueba si P1 nuevo es mayor q el almacenado como tal
     If CSng(fila_puntuaciones.Row.ItemArray(13)) > CSng(matriz_c_e_P1(5, contador_2)) Then
    'Si es asi sustituye al anterior existente en la matriz para ese valor_columna y estrategia
     matriz_c_e_P1(5, contador_2) = fila_puntuaciones.Row.ItemArray(13)
     End If
                        ''''''''
                        'P3_MIN'
                        ''''''''
     'Si el nuevo valor es menor que el almacenado
     If CSng(fila_puntuaciones.Row.ItemArray(14)) < CSng(matriz_c_e_P1(6, contador_2)) Then
    'Si es menor almacena este nuevo valor P1 susituyendo al anterior existente en
    'matriz para ese valor_columna y estrategia
    matriz_c_e_P1(6, contador_2) = fila_puntuaciones.Row.ItemArray(14)
    End If
                        ''''''''
                        'P3_MAX'
                        ''''''''
    'Comprueba si P1 nuevo es mayor q el almacenado como tal
     If CSng(fila_puntuaciones.Row.ItemArray(14)) > CSng(matriz_c_e_P1(7, contador_2)) Then
      'Si es asi sustituye al anterior existente en la                                'matriz para
      ese valor_columna y estrategia
     matriz_c_e_P1(7, contador_2) = fila_puntuaciones.Row.ItemArray(14)
     End If
                         ''''''''
                        'P4_MIN'
                        ''''''''
     'Si el nuevo valor es menor que el almacenado
     If CSng(fila_puntuaciones.Row.ItemArray(15)) < CSng(matriz_c_e_P1(8, contador_2)) Then
    'Si es menor almacena este nuevo valor P1 susituyendo al anterior existente en
    'matriz para ese valor_columna y estrategia
     matriz_c_e_P1(8, contador_2) = fila_puntuaciones.Row.ItemArray(15)
    End If
                        ''''''''
                        'P4_MAX'
                        ''''''''
    'Comprueba si P4 nuevo es mayor q el almacenado como tal
     If CSng(fila_puntuaciones.Row.ItemArray(15)) > CSng(matriz_c_e_P1(9, contador_2)) Then
     'Si es asi sustituye al anterior existente en la matriz para ese valor_columna y estrategia
    matriz_c_e_P1(9, contador_2) = fila_puntuaciones.Row.ItemArray(15)
    End If
                        ''''''''
                        'P5_MIN'
                        ''''''''
    'Si el nuevo valor es menor que el almacenado
If CSng(fila_puntuaciones.Row.ItemArray(16)) < CSng(matriz_c_e_P1(10, contador_2))
Then
'Si es menor almacena este nuevo valor P5 susituyendo al anterior existente en
'matriz para ese valor_columna y estrategia
matriz_c_e_P1(10, contador_2) = fila_puntuaciones.Row.ItemArray(16)
End If
                        ''''''''
                        'P5_MAX'
                        ''''''''
Comprueba si P5 nuevo es mayor q el almacenado como tal
If CSng(fila_puntuaciones.Row.ItemArray(16)) > CSng(matriz_c_e_P1(11, contador_2))
Then
'Si es asi sustituye al anterior existente en la matriz para ese valor_columna y estrategia
 matriz_c_e_P1(11, contador_2) = fila_puntuaciones.Row.ItemArray(16)
 End If
                     aviso_encontrado = True
                    Else
                     contador_2 = contador_2 + 1
                    End If
                Else
                    contador_2 = contador_2 + 1
                End If
            End While

            If aviso_encontrado = False Then
                ReDim Preserve matriz_c_e_P1(11, contador_2)
                matriz_c_e_P1(0, contador_2) = fila_puntuaciones.Row.ItemArray(1)
                matriz_c_e_P1(1, contador_2) = fila_puntuaciones.Row.ItemArray(2)
                matriz_c_e_P1(2, contador_2) = fila_puntuaciones.Row.ItemArray(12)
```

```
                    matriz_c_e_P1(3, contador_2) = fila_puntuaciones.Row.ItemArray(12)
                    matriz_c_e_P1(4, contador_2) = fila_puntuaciones.Row.ItemArray(13)
                    matriz_c_e_P1(5, contador_2) = fila_puntuaciones.Row.ItemArray(13)
                    matriz_c_e_P1(6, contador_2) = fila_puntuaciones.Row.ItemArray(14)
                    matriz_c_e_P1(7, contador_2) = fila_puntuaciones.Row.ItemArray(14)
                    matriz_c_e_P1(8, contador_2) = fila_puntuaciones.Row.ItemArray(15)
                    matriz_c_e_P1(9, contador_2) = fila_puntuaciones.Row.ItemArray(15)
                    matriz_c_e_P1(10, contador_2) = fila_puntuaciones.Row.ItemArray(16)
                    matriz_c_e_P1(11, contador_2) = fila_puntuaciones.Row.ItemArray(16)
                End If
            Next

     'Se rellena la tabla de Puntuaciones min y Max a partir de la matriz matriz_c_e_p1
         For i = 0 To matriz_c_e_P1.GetLength(1) - 1
   registro_puntuaciones_min_max.puntuaciones_min_max_circuito = matriz_c_e_P1(0, i)
   registro_puntuaciones_min_max.puntuaciones_min_max_estrategia = matriz_c_e_P1(1, i)
   registro_puntuaciones_min_max.puntuaciones_min_max_P1_min = CSng(matriz_c_e_P1(2, i))
   registro_puntuaciones_min_max.puntuaciones_min_max_P1_max = CSng(matriz_c_e_P1(3, i))
   registro_puntuaciones_min_max.puntuaciones_min_max_P2_min = CSng(matriz_c_e_P1(4, i))
   registro_puntuaciones_min_max.puntuaciones_min_max_P2_max = CSng(matriz_c_e_P1(5, i))
   registro_puntuaciones_min_max.puntuaciones_min_max_P3_min = CSng(matriz_c_e_P1(6, i))
   registro_puntuaciones_min_max.puntuaciones_min_max_P3_max = CSng(matriz_c_e_P1(7, i))
   registro_puntuaciones_min_max.puntuaciones_min_max_P4_min = CSng(matriz_c_e_P1(8, i))
   registro_puntuaciones_min_max.puntuaciones_min_max_P4_max = CSng(matriz_c_e_P1(9, i))
  registro_puntuaciones_min_max.puntuaciones_min_max_P5_min = CSng(matriz_c_e_P1(10, i))
  registro_puntuaciones_min_max.puntuaciones_min_max_P5_max = CSng(matriz_c_e_P1(11, i))

Puntuaciones_min_maxTableAdapter.Insert(registro_puntuaciones_min_max.puntuaciones_min_max_circuito,
registro_puntuaciones_min_max.puntuaciones_min_max_estrategia, _
registro_puntuaciones_min_max.puntuaciones_min_max_P1_min, _
registro_puntuaciones_min_max.puntuaciones_min_max_P1_max, _
registro_puntuaciones_min_max.puntuaciones_min_max_P2_min, _
registro_puntuaciones_min_max.puntuaciones_min_max_P2_max, _
registro_puntuaciones_min_max.puntuaciones_min_max_P3_min, _
registro_puntuaciones_min_max.puntuaciones_min_max_P3_max, _
registro_puntuaciones_min_max.puntuaciones_min_max_P4_min, _
registro_puntuaciones_min_max.puntuaciones_min_max_P4_max, _
registro_puntuaciones_min_max.puntuaciones_min_max_P5_min, _
registro_puntuaciones_min_max.puntuaciones_min_max_P5_max)
         Next
        Else
            'Si la tabla puntuaciones no tiene ningun registro, concluir la aplicación
            MsgBox("No hay registros válidos para puntuar las FPGAs")
            Application.Exit()
        End If
        Me.Puntuaciones_min_maxTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Puntuaciones_min_max)

        ''''''''''''''''''''''''''
        'PUNTUACIONES NORMALIZADAS'
        ''''''''''''''''''''''''''
 'Se realiza la conversión de cada puntuacion (P1,P2,P3,P4 y P5) de cada Circ,Est y Tec
 'con los  m y b correspondientes. Se agrega a la tabla Puntuaciones_normalizadas
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        Dim registro_normalizados As Normalizados
        Dim m1 As Single
        Dim b1 As Single
        Dim m2 As Single
        Dim b2 As Single
        Dim m3 As Single
        Dim b3 As Single
        Dim m4 As Single
        Dim b4 As Single
        Dim m5 As Single
        Dim b5 As Single
        Dim P1_N As Single
        Dim P2_N As Single
        Dim P3_N As Single
        Dim P4_N As Single
        Dim P5_N As Single
        Dim z As Integer

        Dim total_puntuaciones_min_max As Integer
        Dim fila_puntuaciones_min_max As DataRowView

        'Se obtiene el total de regsitros de la tabla TOTAL_PUNTUACIONES_MIN_MAX
        total_puntuaciones_min_max = Puntuaciones_min_maxBindingSource.Count

        'Para todos los registro de la tabla PUNTUACIONES
        For i = 0 To total_puntuaciones - 1
            fila_puntuaciones = PuntuacionesBindingSource.Item(i)
            z = 0
            aviso_encontrado = False

    'Por cada registro de la tabla Putuaciones, se recorre la tabla puntuAciones_min_maxhasta
coincidir circuito y estrategia. En ese momento se realiza el cociente y se inserta como nuevo
registro de la tabla puntuaciones_normalizados. Nuevamente se usa la bandera de aviso_encontrado para
```

```
salir del while, junto con el total de registros de la misma
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
            While (z < total_puntuaciones_min_max And aviso_encontrado = False)
                fila_puntuaciones_min_max = Puntuaciones_min_maxBindingSource.Item(z)
                'comprueba que es el mismo circutito
If fila_puntuaciones.Row.ItemArray(1) = fila_puntuaciones_min_max.Row.ItemArray(1)
Then
  'Si es el mismo circuito comprueba que es la misma estrategia
If fila_puntuaciones.Row.ItemArray(2) = fila_puntuaciones_min_max.Row.ItemArray(2)
Then
    'Calculo de m y b
                        m1 = (9 / (fila_puntuaciones_min_max.Row.ItemArray(4) -
fila_puntuaciones_min_max.Row.ItemArray(3)))
                        b1 = (fila_puntuaciones_min_max.Row.ItemArray(4) - 10 *
(fila_puntuaciones_min_max.Row.ItemArray(3))) / _
                        ((fila_puntuaciones_min_max.Row.ItemArray(4) -
fila_puntuaciones_min_max.Row.ItemArray(3)))

                        m2 = 9 / (fila_puntuaciones_min_max.Row.ItemArray(6) -
fila_puntuaciones_min_max.Row.ItemArray(5))
                        b2 = (fila_puntuaciones_min_max.Row.ItemArray(6) - 10 *
(fila_puntuaciones_min_max.Row.ItemArray(5))) / _
                        ((fila_puntuaciones_min_max.Row.ItemArray(6) -
fila_puntuaciones_min_max.Row.ItemArray(5)))

                        m3 = 9 / (fila_puntuaciones_min_max.Row.ItemArray(8) -
fila_puntuaciones_min_max.Row.ItemArray(7))
                        b3 = (fila_puntuaciones_min_max.Row.ItemArray(8) - 10 *
(fila_puntuaciones_min_max.Row.ItemArray(7))) / _
                        ((fila_puntuaciones_min_max.Row.ItemArray(8) -
fila_puntuaciones_min_max.Row.ItemArray(7)))

                        m4 = 9 / (fila_puntuaciones_min_max.Row.ItemArray(10) -
fila_puntuaciones_min_max.Row.ItemArray(9))
                        b4 = (fila_puntuaciones_min_max.Row.ItemArray(10) - 10 *
(fila_puntuaciones_min_max.Row.ItemArray(9))) / _
                        ((fila_puntuaciones_min_max.Row.ItemArray(10) -
fila_puntuaciones_min_max.Row.ItemArray(9)))

                        m5 = 9 / (fila_puntuaciones_min_max.Row.ItemArray(12) -
fila_puntuaciones_min_max.Row.ItemArray(11))
                        b5 = (fila_puntuaciones_min_max.Row.ItemArray(12) - 10 *
(fila_puntuaciones_min_max.Row.ItemArray(11))) / _
                        ((fila_puntuaciones_min_max.Row.ItemArray(12) -
fila_puntuaciones_min_max.Row.ItemArray(11)))

                        'Calculo de los valores normalizados segun la ecuación y=mx+b
                        P1_N = fila_puntuaciones.Row.ItemArray(12) * m1 + b1
                        P2_N = fila_puntuaciones.Row.ItemArray(13) * m2 + b2
                        P3_N = fila_puntuaciones.Row.ItemArray(14) * m3 + b3
                        P4_N = fila_puntuaciones.Row.ItemArray(15) * m4 + b4
                        P5_N = fila_puntuaciones.Row.ItemArray(16) * m5 + b5

                        aviso_encontrado = True
                    Else
'si no es es la misma ESTRATEGIA aviso_encontrado = FALSE y sigue buscando,avanza una 'posicion en la
tabla de puntuaciones_min_max y empieza en el while otra vez
                        z = z + 1
                    End If
                Else
'si no es es el mismo CIRCUITO aviso_encotrando seguirá siendo FALSE y sigue buscando,
'avanza uan posicion en la tabla de puntuaciones_min_max y empieza en el while otra vez
                    z = z + 1
                End If
            End While

            'Se inserta el registro en la tabla puntuaciones_normalizadas
      registro_normalizados.normalizados_circuito = fila_puntuaciones.Row.ItemArray(1)
      registro_normalizados.normalizados_estrategia = fila_puntuaciones.Row.ItemArray(2)
      registro_normalizados.normalizados_tecnologia = fila_puntuaciones.Row.ItemArray(3)
      registro_normalizados.normalizados_P1_N = P1_N
      registro_normalizados.normalizados_P2_N = P2_N
      registro_normalizados.normalizados_P3_N = P3_N
      registro_normalizados.normalizados_P4_N = P4_N
      registro_normalizados.normalizados_P5_N = P5_N
      NormalizadosTableAdapter.Insert(registro_normalizados.normalizados_circuito, _
                            registro_normalizados.normalizados_estrategia, _
                            registro_normalizados.normalizados_tecnologia, _
                            registro_normalizados.normalizados_P1_N, _
                            registro_normalizados.normalizados_P2_N, _
                            registro_normalizados.normalizados_P3_N, _
                            registro_normalizados.normalizados_P4_N, _
                            registro_normalizados.normalizados_P5_N)
        Next

        Me.NormalizadosTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Normalizados)
```

```
        TextBox2.Text = NormalizadosBindingSource.Count
    End Sub

    Private Function busqueda_precio(ByVal tecnologia_buscada As String, ByVal fila_tabla As
DataRowView, ByVal tamaño_tabla_precios As Integer)
        Dim precio As Single
        Dim contador As Integer

While ((tecnologia_buscada <> fila_tabla.Row.ItemArray(1)) And contador < tamaño_tabla_precios - 1)
contador = contador + 1
fila_tabla = CaracteristicasBindingSource.Item(contador)
End While
        If tecnologia_buscada = fila_tabla.Row.ItemArray(1) Then
            precio = fila_tabla.Row.ItemArray(2)
        Else
            precio = 0
        End If

        Return precio
    End Function

    Private Function conjunto_valor_columna(ByVal tabla As BindingSource, ByVal columna As Integer)
'Devuelve un array que contiene los elementos, sin repetir, de la columna 1 de la tabla especificada
como parámetro.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        Dim fila As DataRowView
        Dim total_tabla As Integer
        Dim array_almacen() As String = {}
        Dim valor_columna As String
        Dim contador_valor_columna As Integer = 0

        total_tabla = tabla.Count

        For i = 0 To total_tabla - 1
            'Se carga una fila de la tabla indicada
            fila = tabla.Item(i)

            'Conversion a mayúsculas
            valor_columna = fila.Row.ItemArray(columna)
            valor_columna = valor_columna.ToUpper

   'Si valor_columna del registro actual no se ha guardado todavia en array_almacen
            'se guarda
            If Not array_almacen.Contains(valor_columna) Then
                ReDim Preserve array_almacen(contador_valor_columna)
                array_almacen(contador_valor_columna) = valor_columna
                contador_valor_columna = contador_valor_columna + 1
            End If
        Next
        Return array_almacen
    End Function

    Private Function conjunto_valor_columna(ByVal tabla As BindingSource, ByVal columna As Integer,
ByVal criterio As String, ByVal columna_filtro As Integer)
    'Devuelve un array que contiene los elementos, sin repetir, de la columna "COLUMNA" de la tabla
especificada como parámetro si coincide con CRITERIO valor perteneciente a otra columna
COLUMNA_FILTRO
        Dim fila As DataRowView
        Dim total_tabla As Integer
        Dim array_almacen() As String = {}
        Dim valor_columna As String
        Dim contador_valor_columna As Integer = 0
        Dim valor_filtro As String

        total_tabla = tabla.Count

        For i = 0 To total_tabla - 1
            'Se carga una fila de la tabla indicada
            fila = tabla.Item(i)

            'Conversion a mayúsculas
            valor_columna = fila.Row.ItemArray(columna)
            valor_columna = valor_columna.ToUpper
            valor_filtro = fila.Row.ItemArray(columna_filtro)
            valor_filtro = valor_filtro.ToUpper
            criterio = criterio.ToUpper

    'Si el valor_columna del registro actual no se ha guardado todavia en array_almacen
    'se guarda si valor_filtro = criterio
            If Not array_almacen.Contains(valor_columna) Then
                If valor_filtro = criterio Then
                    ReDim Preserve array_almacen(contador_valor_columna)
                    array_almacen(contador_valor_columna) = valor_columna
                    contador_valor_columna = contador_valor_columna + 1
                End If
            End If
```

```
        Next
        Return array_almacen
    End Function

    Private Sub ToolStripButton4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ToolStripButton4.Click
        Application.Exit()
    End Sub

    Private Sub PuntuacionesToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PuntuacionesToolStripMenuItem.Click
        Me.PuntuacionesTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Puntuaciones)
        Me.NormalizadosTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Normalizados)
Me.Fallos_puntuacionTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Fallos_puntuacion)

        TextBox1.Text = PuntuacionesBindingSource.Count
        TextBox2.Text = NormalizadosBindingSource.Count
        TextBox3.Text = Fallos_puntuacionBindingSource.Count
    End Sub

    Private Sub LecturaDeInformesToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles LecturaDeInformesToolStripMenuItem.Click
        Me.Hide()
        Form1.Show()
    End Sub

    Private Sub CircuitosImplementadosToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles CircuitosImplementadosToolStripMenuItem.Click
        Form3.Show()
        Me.Hide()
    End Sub

    Private Sub CircuitosPorSectorToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CircuitosPorSectorToolStripMenuItem.Click
        Form5.Show()
        Me.Hide()
    End Sub
End Class
```

FORM3.vb

```
Public Class Form3
    Structure tec_circ
        Public tecnologia As String
        Public circuitos As String
    End Structure


    Private Sub Form3_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    MyBase.Load
         Me.NormalizadosTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Normalizados)
Me.Circuitos_implementadosTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Circuitos_implementados)
        TextBox1.Text = Circuitos_implementadosBindingSource.Count
    End Sub

    Public Function indice_en_array(ByVal elemento As String, ByVal array As String())
        ' Determina el indice del elemento del array que coincide con el "elemento" buscado
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        Dim A As Integer = 0
        Dim indice As Integer

        For A = 0 To array.Length - 1
            If elemento = array(A) Then
                indice = A
                Exit For
            Else
                'Si el elemento no está en el array devolver un "indice" -1
                indice = -1
            End If
        Next A
        Return indice
    End Function

    Private Function array_circuitos_por_tecnologia(ByVal tecnologia As String, ByVal estrategia As
String, ByVal tabla As BindingSource)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Devuelve un array cuyos elementos son las puntuaciones Px_N (P1_N ó P2_N ó etc) de una determinada
tecnologia referida como argumento. Ademas, solo devuelve el Px_N si coincide con la estrategia
especificada en el parámetro del mismo nombre. El último parámtero hace referencia a la tabla donde
debe buscarse las Px_N que se devolveran como array.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        Dim total_tabla As Integer
        Dim fila_tabla As DataRowView
```

```
        Dim cont_circ As Integer = 0
        Dim array_circuitos() As String = {}

        estrategia = estrategia.ToUpper
        total_tabla = tabla.Count

        'Recorre toda la tabla TABLA
        For i = 0 To total_tabla - 1
            fila_tabla = tabla.Item(i)
    'si el campo tecnologia coincide con el parametro TECNOLOGIA y la estrategia tambien
        If ((fila_tabla.Row.ItemArray(3) = tecnologia) And (fila_tabla.Row.ItemArray(2) =
        estrategia)) Then
                'guarda el circuito en el array
                ReDim Preserve array_circuitos(cont_circ)
                array_circuitos(cont_circ) = fila_tabla.Row.ItemArray(1)
                'Se incrementa el contador que dimensiona este array de puntuaciones
                cont_circ = cont_circ + 1
         End If
        Next
        'Devuelve un array cuyos elementos son los circuitos implementados por cada tecnologia
        Return array_circuitos
    End Function

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Application.Exit()
    End Sub

    Private Sub PuntuacionesToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles PuntuacionesToolStripMenuItem.Click
        'Puntuaciones
        ''''''''''''''''''''
        Form2.Show()
        Me.Hide()
    End Sub

    Private Sub LecturaDeInformesToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles LecturaDeInformesToolStripMenuItem.Click
        'Lectura de informes
        ''''''''''''''''''''
        Form1.Show()
        Me.Hide()
    End Sub

    Private Sub CircuitosImplementadosToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles CircuitosImplementadosToolStripMenuItem.Click
        'Circuitos implementados
        ''''''''''''''''''''''''
     Me.Circuitos_implementadosTableAdapter.Fill(Resultados_benchamrkDataSet.Circuitos_implementados)
        TextBox1.Text = Circuitos_implementadosBindingSource.Count
    End Sub

    Private Sub ToolStripButton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles ToolStripButton1.Click
        Me.Circuitos_implementadosTableAdapter.DeleteQuery()

        'Nuevo
        ''''''
        Dim total As Integer = 0
        Dim total_normalizados As Integer
        Dim fila_normalizadas As DataRowView
        Dim tecnologia As String
        Dim tec_temp() As String = {}
        Dim cont_tec As Integer = 0
        Dim array_circuitos() As String = {}
        Dim tecnologia_circuitos As tec_circ


        total_normalizados = NormalizadosBindingSource.Count

        '''''''''''''''''''''''''''''''''
        ' TABLA CIRCUITOS_IMPLEMENTADOS '
        '''''''''''''''''''''''''''''''''
'Se va leyendo cada registro de la tabla NORMALIZADOS para detectar todos los registros que lleven la
misma tecnología(y estrategia), se anota el circuito para tener la lista de los circuitos
implementados en cada FPGA
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        While total < total_normalizados - 1
            fila_normalizadas = NormalizadosBindingSource.Item(total)
            tecnologia = fila_normalizadas.Row.ItemArray(3)
            'Si la tecnologia del registro que se está leyendo no ha sido aún analizada,
            'esto es, esté almacenada en el array TEC_TEMP, se analizará, si no, se lee
            'el registro siguiente
            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
            If tec_temp.Contains(tecnologia) Then
                'La tecnologia del registro actual ya fue leida, pasamos el registro siguiente
                total = total + 1
```

```
            Else
                'No leida aún
                '''''''''''''
'Se llama a una función que devuelve un array con los circuitos implementados para esa
tecnologia

 array_circuitos = array_circuitos_por_tecnologia(tecnologia, "balanced", NormalizadosBindingSource)
                Dim total2 As Integer
                total2 = array_circuitos.Count

                If total2 <> 0 Then
                  'Para la tecnologia del registro analizado se obtiene el valor de todos sus campos
                    tecnologia_circuitos.tecnologia = tecnologia
                    tecnologia_circuitos.circuitos = Join(array_circuitos, ",")

                    'Insertar en las tablas
Circuitos_implementadosTableAdapter.Insert(tecnologia_circuitos.tecnologia,tecnologia_circuitos.circuitos)
                End If

'Se almacena en un array auxiliar las tecnologías que ya han sido revisadas, para que no se repita
el proceso cuando vuelva a encontrarla en otro registro
                ReDim Preserve tec_temp(cont_tec)
                tec_temp(cont_tec) = tecnologia
                cont_tec = cont_tec + 1
                total = total + 1
            End If
        End While

Me.Circuitos_implementadosTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Circuitos_implementados)
        TextBox1.Text = Circuitos_implementadosBindingSource.Count
    End Sub

    Private Sub ToolStripButton3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles ToolStripButton3.Click
        Form4.Show()
        Me.Hide()
    End Sub

   Private Sub ToolStripButton4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
   Handles ToolStripButton4.Click
        Application.Exit()
   End Sub

    Private Sub CircuitosPorSectorToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles CircuitosPorSectorToolStripMenuItem.Click
        Form5.Show()
        Me.Hide()
    End Sub
End Class
```

FORM4.vb

```
Public Class Form4

Private Sub Form4_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        ListBox1.SelectedIndex = -1
If (ListBox2.SelectedIndex = -1 And ListBox3.SelectedIndex = -1 And ListBox4.SelectedIndex = -1)
Then Button7.Enabled = False
Else Button7.Enabled = True
End If
        Me.CircuitoTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Circuito)
        TextBox1.Text = CircuitoBindingSource.Count
        Button4.Enabled = False
        Button5.Enabled = False
        Button6.Enabled = False
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

        'AÑADIR A BENCH_AUTO'
        '''''''''''''''''''''
        ListBox2.Enabled = True
        Button4.Enabled = True

         'Dado que los datos proceden de la tabla CIRCUITO las lineas que aparecen el listbox1 son
         filas de la tabla (datarowview).De esta fila (circuito_seleccionado) se extraerá solo el
         campo circuito
         Dim circuito_seleccionado As DataRowView
        Dim array_circuitos_seleccionados(ListBox1.SelectedItems.Count - 1) As String

        If ListBox1.SelectedIndex <> -1 Then
```

```
        For i As Integer = 0 To Me.ListBox1.SelectedItems.Count - 1
            circuito_seleccionado = ListBox1.SelectedItems(i)
            array_circuitos_seleccionados(i) = circuito_seleccionado.Row.ItemArray(1)
        Next
    Else
        MsgBox("No ha seleccionado ningún circuito")
    End If

    For i = 0 To array_circuitos_seleccionados.Count - 1
        If Not ListBox2.Items.Contains(array_circuitos_seleccionados(i)) Then
            ListBox2.Items.Add(array_circuitos_seleccionados(i))
        End If
    Next

    TextBox2.Text = ListBox2.Items.Count
    ListBox1.SelectedIndex = -1
    Button7.Enabled = True
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click

    'AÑADIR A BENCH_CONSUMO'
    '''''''''''''''''''''''
    ListBox3.Enabled = True
    Button5.Enabled = True

    'Dado que los datos proceden realmentes de la tabla CIRCUITO
    'las lineas que aparecen el listbox1, son filas de la tabla (datarowview)
    'De esta fila (circuito_seleccionado) se extraerá solo el campo que interese, el circuito
    Dim circuito_seleccionado As DataRowView
    Dim array_circuitos_seleccionados(ListBox1.SelectedItems.Count - 1) As String

    If ListBox1.SelectedIndex <> -1 Then
        For i As Integer = 0 To Me.ListBox1.SelectedItems.Count - 1
            circuito_seleccionado = ListBox1.SelectedItems(i)
            array_circuitos_seleccionados(i) = circuito_seleccionado.Row.ItemArray(1)
        Next
        TextBox3.Text = array_circuitos_seleccionados.Count
    Else
        MsgBox("No ha seleccionado ningún circuito")
    End If

    For i = 0 To array_circuitos_seleccionados.Count - 1
        If Not ListBox3.Items.Contains(array_circuitos_seleccionados(i)) Then
            ListBox3.Items.Add(array_circuitos_seleccionados(i))
        End If
    Next

    ListBox1.SelectedIndex = -1
    Button7.Enabled = True
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button3.Click

    'AÑADIR A BENCH_COMUNICACIONES'
    '''''''''''''''''''''''''''''''''
    ListBox4.Enabled = True
    Button6.Enabled = True

    'Dado que los datos proceden realmentes de la tabla CIRCUITO
    'las lineas que aparecen el listbox1, son filas de la tabla (datarowview)
    'De esta fila (circuito_seleccionado) se extraerá solo el campo que interese, circuito
    Dim circuito_seleccionado As DataRowView
    Dim array_circuitos_seleccionados(ListBox1.SelectedItems.Count - 1) As String

    If ListBox1.SelectedIndex <> -1 Then
        For i As Integer = 0 To Me.ListBox1.SelectedItems.Count - 1
            circuito_seleccionado = ListBox1.SelectedItems(i)
            array_circuitos_seleccionados(i) = circuito_seleccionado.Row.ItemArray(1)
        Next
    Else
        MsgBox("No ha seleccionado ningún circuito")
    End If

    For i = 0 To array_circuitos_seleccionados.Count - 1
        If Not ListBox4.Items.Contains(array_circuitos_seleccionados(i)) Then
            ListBox4.Items.Add(array_circuitos_seleccionados(i))
        End If
    Next
    TextBox4.Text = ListBox4.Items.Count
    ListBox1.SelectedIndex = -1
    Button7.Enabled = True
End Sub
```

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button4.Click
    'BORRAR auto'
    '''''''''''''''
    Button7.Enabled = False

    Dim array_indices_sel() As Integer = {}
    Dim cont As Integer = 0
    Dim total_items As Integer
    total_items = ListBox2.Items.Count
    For i = 0 To ListBox2.Items.Count - 1
        If ListBox2.GetSelected(i) = True Then
            ReDim Preserve array_indices_sel(cont)
            array_indices_sel(cont) = i
            cont = cont + 1
        End If
    Next
    Dim t_sel As Integer
    t_sel = array_indices_sel.Count
    For i = array_indices_sel.Count - 1 To 0 Step -1
        ListBox2.Items.RemoveAt(array_indices_sel(i))
    Next
    TextBox2.Text = ListBox2.Items.Count
End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button5.Click
    'BORRAR CONSUMO'
    '''''''''''''''
    Button7.Enabled = False

    Dim array_indices_sel() As Integer = {}
    Dim cont As Integer = 0
    Dim total_items As Integer
    total_items = ListBox3.Items.Count
    For i = 0 To ListBox3.Items.Count - 1
        If ListBox3.GetSelected(i) = True Then
            ReDim Preserve array_indices_sel(cont)
            array_indices_sel(cont) = i
            cont = cont + 1
        End If
    Next
    Dim t_sel As Integer
    t_sel = array_indices_sel.Count
    For i = array_indices_sel.Count - 1 To 0 Step -1
        ListBox3.Items.RemoveAt(array_indices_sel(i))
    Next
    TextBox3.Text = ListBox3.Items.Count
End Sub

Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button6.Click
    'BORRAR COMUNICACIONES'
    '''''''''''''''''''''''
    Button7.Enabled = False

    Dim array_indices_sel() As Integer = {}
    Dim cont As Integer = 0
    Dim total_items As Integer
    total_items = ListBox4.Items.Count
    For i = 0 To ListBox4.Items.Count - 1
        If ListBox4.GetSelected(i) = True Then
            ReDim Preserve array_indices_sel(cont)
            array_indices_sel(cont) = i
            cont = cont + 1
        End If
    Next
    Dim t_sel As Integer
    t_sel = array_indices_sel.Count
    For i = array_indices_sel.Count - 1 To 0 Step -1
        ListBox4.Items.RemoveAt(array_indices_sel(i))
    Next
    TextBox4.Text = ListBox4.Items.Count
End Sub

Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button8.Click
    Form5.CircuitosPorSectorToolStripMenuItem.PerformClick()
    Form5.Show()
    Me.Hide()
End Sub

Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button7.Click
    '''''''''''
    ' ENVIAR '
```

```
        ''''''''''
        'Inserta en cada tabla correspondiente de la BD el total de circuito seleccionado
        'por cada sector
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        Button7.Enabled = True
        Bench_autoTableAdapter.DeleteQuery()
        Bench_comunicacionesTableAdapter.DeleteQuery()
        Bench_consumoTableAdapter.DeleteQuery()

        For i = 0 To ListBox2.Items.Count - 1
            Bench_autoTableAdapter.Insert(ListBox2.Items(i))
        Next

        For i = 0 To ListBox3.Items.Count - 1
            Bench_consumoTableAdapter.Insert(ListBox3.Items(i))
        Next

        For i = 0 To ListBox4.Items.Count - 1
            Bench_comunicacionesTableAdapter.Insert(ListBox4.Items(i))
        Next

        MsgBox("Las tablas de la BD han sido actualizadas")
    End Sub
End Class
```

FORM5.vb

```
Public Class Form5

    Private Sub Form5_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    MyBase.Load
        Bench_autoDataGridView.Refresh()

        Me.Bench_consumoTableAdapter.Fill(Me.Resultados_benchamrkDataSet.bench_consumo)
        Me.Bench_comunicacionesTableAdapter.Fill(Me.Resultados_benchamrkDataSet.bench_comunicaciones)
        Me.Bench_autoTableAdapter.Fill(Me.Resultados_benchamrkDataSet.bench_auto)

        TextBox1.Text = Bench_autoBindingSource.Count
        TextBox2.Text = Bench_consumoBindingSource.Count
        TextBox3.Text = Bench_comunicacionesBindingSource.Count
    End Sub

    Private Sub CircuitosImplementadosToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles CircuitosImplementadosToolStripMenuItem.Click
        Form3.Show()
        Me.Hide()
    End Sub

    Private Sub LecturaDeInformesToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles LecturaDeInformesToolStripMenuItem.Click
        Form1.Show()
        Me.Hide()
    End Sub

    Private Sub PuntuacionesToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles PuntuacionesToolStripMenuItem.Click
        Form2.Show()
        Me.Hide()
    End Sub

    Private Sub CircuitosPorSectorToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles CircuitosPorSectorToolStripMenuItem.Click
        Bench_autoTableAdapter.Fill(Resultados_benchamrkDataSet.bench_auto)
        Bench_comunicacionesTableAdapter.Fill(Resultados_benchamrkDataSet.bench_comunicaciones)
        Bench_consumoTableAdapter.Fill(Resultados_benchamrkDataSet.bench_consumo)
        TextBox1.Text = Bench_autoBindingSource.Count
        TextBox2.Text = Bench_consumoBindingSource.Count
        TextBox3.Text = Bench_comunicacionesBindingSource.Count
    End Sub

    Private Sub ToolStripButton4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles ToolStripButton4.Click
        Application.Exit()
    End Sub

    Private Sub ToolStripButton3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles ToolStripButton3.Click
        Form4.Show()
        Me.Hide()
    End Sub
End Class
```

Anexo B.2. AsistenteFPGA

Form1.vb

```
Public Class Form1

    Public array_seleccionados() As String

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load

        'Se cargan las tablas que proporcionaran valores a las listas desplegables

        Me.CircuitoTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Circuito)
        Me.EncapsuladosTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Encapsulados)
        Me.PinesTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Pines)
        Me.Unidades_logicasTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Unidades_logicas)
        Me.Temp_maxTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Temp_max)
        Me.Temp_minTableAdapter.Fill(Resultados_benchamrkDataSet.Temp_min)
        Me.PrecioTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Precio)

        'Se cargan las tablas que guardan los circuitos que podran seleccionarse por sector
        Me.Bench_consumoTableAdapter.Fill(Me.Resultados_benchamrkDataSet.bench_consumo)
        Me.Bench_comunicacionesTableAdapter.Fill(Me.Resultados_benchamrkDataSet.bench_comunicaciones)
        Me.Bench_autoTableAdapter.Fill(Me.Resultados_benchamrkDataSet.bench_auto)

        'Reinicio de la pantalla
        Button1.Enabled = True
        Button2.Enabled = False
        Button3.Enabled = False
        Button4.Enabled = False

        ComboBox1.SelectedIndex = 0
        ComboBox2.SelectedIndex = 0
        ComboBox3.SelectedIndex = 0
        ComboBox4.SelectedIndex = 0
        ComboBox5.SelectedIndex = 0
        ComboBox6.SelectedIndex = 0
        ComboBox10.SelectedIndex = 0
        ComboBox12.SelectedIndex = 0

        TextBox1.Text = ""
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    Button1.Click
        'Botón SIGUIENTE'
        '''''''''''''''''
        'Pasa al siguiente formulario que calculará y mostrará los resultados
        '-------------------------------------------------------------------
        Tabla_seleccion_FPGATableAdapter.DeleteQuery()

        Button1.Enabled = False
        Button2.Enabled = False

        Form2.Show()
        Me.Close()
    End Sub

    Private Sub ToolStripButton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles ToolStripButton1.Click
        Application.Exit()
    End Sub

    Private Sub ComboBox12_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs)
    Handles ComboBox12.SelectedIndexChanged
        ListBox1.Items.Clear()
        Button2.Enabled = False
        Button3.Enabled = False
        TextBox1.Text = ""

        'Solo en el caso de solicitar seleccionar circuitos se activará el botón GENERAR CIRCUITOS
        Select Case ComboBox12.Text
            Case "Circuitos del PdC seleccionados de una lista"
                Button4.Enabled = True
            Case "Todas las FPGAs con circuitos comunes"
                Button4.Enabled = False
```

```
        Case "FPGAs con todos los circuitos"
            Button4.Enabled = False
    End Select
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
    'Boton AÑADIR
    '''''''''''''
    'Dado que los datos proceden realmentes de la tabla CIRCUITO
    'las lineas que aparecen el listbox1, son filas de una tabla (datarowview)
    'De esta fila (circuito_seleccionado) se extraerá solo el campo que interese, circuito en si
    '´-----------------------------------------------------------------------------------------------
    Dim array_circuitos_seleccionados(ListBox1.SelectedItems.Count - 1) As String
    Dim circuito_seleccionado As String

    Button2.Enabled = False

    'Si existe algún elemento seleccionado en el listobox (<>-1) se procede a almacenar
    'los items elegidos en el array correspondiente
    '-------------------------------------------------------------------------------------
    If ListBox1.SelectedIndex <> -1 Then
        For i As Integer = 0 To Me.ListBox1.SelectedItems.Count - 1
            circuito_seleccionado = ListBox1.SelectedItems(i)
            'array_circuitos_seleccionados(i) = circuito_seleccionado.Row.ItemArray(1)
            array_circuitos_seleccionados(i) = circuito_seleccionado
        Next
       'El array se transfiere a otro de carácter público para ser utilizado en otros formularios
        '-------------------------------------------------------------------------------------------
        array_seleccionados = array_circuitos_seleccionados

        TextBox1.Text = Join(array_circuitos_seleccionados, " , ")
        TextBox2.Text = array_circuitos_seleccionados.Count
    Else
        TextBox1.Text = "No ha seleccionado ningún circuito"
    End If

    'Deselecciona los circuitos del listbox y habilita el botón BOrrar
    '-----------------------------------------------------------------
    ListBox1.SelectedIndex = -1
    Button3.Enabled = True
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button3.Click
    'botón Borrar
    '''''''''''''
    Button3.Enabled = False
    If Not array_seleccionados Is Nothing Then
        Array.Clear(array_seleccionados, 0, array_seleccionados.Count)
    End If
    TextBox1.Text = ""
    TextBox2.Text = ""
    Button2.Enabled = True
End Sub

Private Sub ComboBox6_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox6.SelectedIndexChanged
    'Selección de sector'
    '''''''''''''''''''''
    TextBox1.Text = ""
    Button2.Enabled = False
    Button3.Enabled = False
    If ComboBox12.SelectedIndex = 2 Then
        ListBox1.Items.Clear()
        Button4.Enabled = True
    End If
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button4.Click
    '''''''''''''''''''''''''''''
    ' GENERAR LISTA DE CIRCUITOS'
    '''''''''''''''''''''''''''''

    Button4.Enabled = False
    Button2.Enabled = True
    TextBox1.Text = ""
    'Se añaden los que correspondan segun sector elegido anteriormente

    If ComboBox12.SelectedIndex = 2 Then '--SELECCIONAR CIRCUITOS
        'TODOS
        If ComboBox6.SelectedIndex = 0 Then
            For i = 0 To CircuitoBindingSource.Count - 1
                Dim a As DataRowView
                a = CircuitoBindingSource.Item(i)
```

```
                    ListBox1.Items.Add(a.Row.ItemArray(1))
                Next
                'AUTO
            ElseIf ComboBox6.SelectedIndex = 1 Then
                For i = 0 To Bench_autoBindingSource.Count - 1
                    Dim a As DataRowView
                    a = Bench_autoBindingSource.Item(i)
                    ListBox1.Items.Add(a.Row.ItemArray(1))
                Next
                'CONSUMO
            ElseIf ComboBox6.SelectedIndex = 2 Then
                For i = 0 To Bench_consumoBindingSource.Count - 1
                    Dim a As DataRowView
                    a = Bench_consumoBindingSource.Item(i)
                    ListBox1.Items.Add(a.Row.ItemArray(1))
                Next
                'COMUNICACIONES
            ElseIf ComboBox6.SelectedIndex = 3 Then
                For i = 0 To Bench_comunicacionesBindingSource.Count - 1
                    Dim a As DataRowView
                    a = Bench_comunicacionesBindingSource.Item(i)
                    ListBox1.Items.Add(a.Row.ItemArray(1))
                Next
            End If
        End If
    End Sub

End Class
```

Form2.VB

```
Public Class Form2
    Structure tabla_seleccion_fpga
        Public tecnologia As String
        Public circuito As String
        Public P1_global As Single
        Public P2_global As Single
        Public P3_global As Single
        Public P4_global As Single
        Public P5_global As Single
        Public P_total As Single
        Public precio As Single
    End Structure

Private Sub ToolStripButton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ToolStripButton1.Click
        'NUEVA BÚSQUEDA'
        '''''''''''''''''
        'Reset de banderas
        viene_de_volver = False
        no_seleccion = False

        'Borra el contenido de las tablas que guardarán los resultadoa
        Tabla_seleccion_FPGATableAdapter.DeleteQuery()
        Tabla_seleccion_FPGA_PFTableAdapter.DeleteQuery()

        'Vuelve al formular de Selección
        Me.Close()
        Form1.Show()
End Sub

Private Sub ToolStripButton2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ToolStripButton2.Click
        'Botón Salir
        Application.Exit()
End Sub

Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
       Me.Tabla_seleccion_FPGATableAdapter.Fill(Me.Resultados_benchamrkDataSet.Tabla_seleccion_FPGA)
       Me.CaracteristicasTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Caracteristicas)
Me.Circuitos_implementadosTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Circuitos_implementados)
        Me.NormalizadosTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Normalizados)
        Me.CircuitoTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Circuito)

        If viene_de_volver = False Then
            'Punto de retorno tras pulsar NUEVA BÚSQUEDA o es la primera vez
```

```
Dim sel_encapsulado As String
Dim sel_pines_minimo As String
Dim sel_pines_min_int As Integer
Dim sel_temp_max As String
Dim sel_temp_max_int As Integer
Dim sel_temp_min As String
Dim sel_temp_min_int As Integer
Dim sel_ul_min As String
Dim sel_ul_min_int As Integer
Dim sel_precio_min As String
Dim sel_precio_min_sng As Single
Dim sel_sector As String


'Se importan los valores seleccionados en la ventana anterior y se convierten
'cada string al tipo adecuado segun parámetro
'Si no se ha seleccionado ningun valor en los parámetros númericos,
'se le asigna el valor máximo o mínimo obtenido en el form Principal
'´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
''''''''''''''
' ENCAPSULADO'
''''''''''''''
Dim sin_encapsulado As Boolean = False
sel_encapsulado = Form1.ComboBox1.Text

'si  no se selecciona ningun encpasulado se supones que sirve cualquiera
'Se avisará de la circunstancia mediante una bandera: sin_encapsulado
'y se listarán las fpga sin filtrar por ningún encapsulado.

If sel_encapsulado = "" Then
    sin_encapsulado = True
Else
    sin_encapsulado = False
End If

'''''''
'PINES'
'''''''
sel_pines_minimo = Form1.ComboBox2.Text
If sel_pines_minimo = "" Then
    sel_pines_min_int = Principal.pines_max
Else
    sel_pines_min_int = CInt(sel_pines_minimo)
End If

'''''''
'T_MAX'
'''''''
sel_temp_max = Form1.ComboBox4.Text
If sel_temp_max = "" Then
    sel_temp_max_int = Principal.tem_max_max
Else
    sel_temp_max_int = CInt(sel_temp_max)
End If

'''''''
'T_MIN'
'''''''
sel_temp_min = Form1.ComboBox5.Text
If sel_temp_min = "" Then
    sel_temp_min_int = Principal.temp_min_min
Else
    sel_temp_min_int = CInt(sel_temp_min)
End If

''''''''
'UL_MIN'
''''''''
sel_ul_min = Form1.ComboBox3.Text
If sel_ul_min = "" Then
    sel_ul_min_int = Principal.ul_max
Else
    sel_ul_min_int = CInt(sel_ul_min)
End If

''''''''
'PRECIO'
''''''''
sel_precio_min = Form1.ComboBox10.Text
If sel_precio_min = "" Then
    sel_precio_min_sng = Principal.precio_max
Else
    sel_precio_min_sng = CSng(sel_precio_min)
End If

''''''''
```

```
            'SECTOR'
            ''''''''
            sel_sector = Form1.ComboBox6.Text

            '-----------------------------------------------------
            ' 1. FPGAS  que cumplen con las condiciones de filtro
            '-----------------------------------------------------
            'Definición de variables para acceder a la tabla CARACTERISTICAS
            '---------------------------------------------------------------
            Dim fila As DataRowView

         'Variable que guardará el nombre de las tecnologias que se ajustan a los criterios elegidos
         '-------------------------------------------------------------------------------------------
            Dim array_fpga_criterios() As String = {}

            'Variables que tomarán el valor de cada campo del registro en uso de CARACTERISTICAS
            '-----------------------------------------------------------------------------------
            Dim cont_fpga_criterios As Integer = 0
            Dim campo_precio As Single
            Dim campo_encapsulado As String
            Dim campo_ul As Integer
            Dim campo_pines As Integer
            Dim campo_t_max As Integer
            Dim campo_t_min As Integer

         'Por cada registro de CARACTERISTICAS, se comprueba qué tecnología cumple todos los filtros
         'establecidos en Form1(Asistente de búsqueda)
         '-------------------------------------------------------------------------------------------
            For i = 0 To CaracteristicasBindingSource.Count - 1
                fila = CaracteristicasBindingSource.Item(i)
                campo_precio = fila.Row.ItemArray(2)
                campo_encapsulado = fila.Row.ItemArray(3)
                campo_ul = CInt(fila.Row.ItemArray(5))
                campo_pines = CInt(fila.Row.ItemArray(8))
                campo_t_max = CInt(fila.Row.ItemArray(13))
                campo_t_min = CInt(fila.Row.ItemArray(14))

                If sin_encapsulado = True Then
                    'No se especificó encapsulado, no se incluye como filtro de la búsqueda
                    If (campo_precio <= sel_precio_min_sng And campo_ul >= sel_ul_min_int And _
                        campo_pines >= sel_pines_min_int And campo_t_max >= sel_temp_max_int And _
                        campo_t_min <= sel_temp_min_int) Then
                        ReDim Preserve array_fpga_criterios(cont_fpga_criterios)
                        array_fpga_criterios(cont_fpga_criterios) = fila.Row.ItemArray(1)
                        cont_fpga_criterios = cont_fpga_criterios + 1
                    End If
                Else
                    'Encapsulado concreto, se incluye en la búsqueda
                    If (campo_precio <= sel_precio_min_sng And campo_encapsulado = sel_encapsulado
And campo_ul >= sel_ul_min_int And campo_pines >= sel_pines_min_int And
campo_t_max >= sel_temp_max_int And campo_t_min <= sel_temp_min_int)
                     Then
                        ReDim Preserve array_fpga_criterios(cont_fpga_criterios)
                        array_fpga_criterios(cont_fpga_criterios) = fila.Row.ItemArray(1)
                        cont_fpga_criterios = cont_fpga_criterios + 1
                    End If
                End If
            Next
            '-------------------------------------------------------------------------------
            'Se comprueba si el filtrado devuelve alguna FPGA que se ajuste a todos los criterios
            'Si es asi sigue adelante, sino muestra la ventana CLASIFICACIÓN pero sin registros
            '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            If cont_fpga_criterios <> 0 Then
                '-----------------------------------------------------
                ' 2. Puntuación de las FPGAs seleccionadas en el punto anterior
                '-----------------------------------------------------
                'SI NO SE SELECCIONA NINGUN SECTOR SE CONSIDERARÁN TODOS LOS CIRCUITOS DEL PdC
               '-------------------------------------------------------------------------------------
                'Estructura que guardará los valores
                Dim tabla_FPGAs As tabla_seleccion_fpga
                Dim matriz_tec_punt(7, 0) As String
                Dim array_indice() As String = {}
                Dim no_seleccion As Boolean = False

                If Form1.ComboBox6.Text = "" Then '--ningun sector
                    Dim todos_circuitos() As String = {}

                    'Llamada a función que convierte elementos de una tabla en un array
                    'Al no haber seleccionado ningún sector, el array de circuitos estará formado por
                    'el total del PdC
                    todos_circuitos = array_desde_tabla(CircuitoBindingSource)

                    ' a) FPGAs con todos los circuitos'
                    '-----------------------------
                 If Form1.ComboBox12.SelectedIndex = 0 Then '--Con todos los circuitos (Todo el PdC)
                    Dim tec_todos_circuitos() As String = {}
```

```
                'Llamada a función que determina qué tecnologias de una lista han implementado todos
                'los circuitos de la otra (TODO EL PDC EN ESTE CASO)
            tec_todos_circuitos = comprueba_tecnologia_circ_imp(array_fpga_criterios, todos_circuitos)

                ' Si existe alguna FPGA de la lista que haya podido implementar todos los circuitos
                'del PdC (lista circuito), se sigue adelante
                        If tec_todos_circuitos.Length <> 0
                         Then
             'Llamada a función que obtiene las puntuaciones por cada tecnologia de lista_tecnologias
se obtiene una matriz con nombre tec,P1,...,P5, PG y lista_circuitos como cadena de texto
                         ReDim Preserve matriz_tec_punt(7, tec_todos_circuitos.Count - 1)
                         matriz_tec_punt = Punt_lista_tec_circ(tec_todos_circuitos, todos_circuitos)

                            ''Muestra resultados
                            ''-----------------------------------------------------------
                            'En todos los apartados se almacenarán los resultados a mostrar
                            'en las mismas variables globales, para hacer la presentación
                            'comun a todas las opciones
                            '-----------------------------------------------------------
                            array_indice = tec_todos_circuitos
                        Else
                            MsgBox("Las FPGAs no tienen implementados circuitos comunes de la lista")
                        End If '--si ninguna FPGAs de la lista ha implementado todos los circuitos

                        'Valores comunes a ambas situaciones para mostrar en pantalla
                        texto_para_text2 = "FPGAs con todos los circuitos implementados"
                        texto_para_text1 = "Global"
                        variable_de_circuitos = todos_circuitos


                        'b) Todas las FPGAs filtradas con los circuitos comunes'
                        '-----------------------------------------------------
                    ElseIf Form1.ComboBox12.SelectedIndex = 1 Then
                        Dim circuitos_comunes_a_todas() As String = {}
                        ReDim Preserve matriz_tec_punt(7, array_fpga_criterios.Count - 1)
   circuitos_comunes_a_todas = todas_tecnologias_circ_comunes(array_fpga_criterios, todos_circuitos)

                        'Si la lista de FPGAs no presenta ningun circuito común
                        If circuitos_comunes_a_todas.Count <> 0 Then
            matriz_tec_punt = Punt_lista_tec_circ(array_fpga_criterios, circuitos_comunes_a_todas)

                            'Muestra resultados
                            '-----------------------------------------------------------
                            array_indice = array_fpga_criterios

                        Else
                            MsgBox("Las FPGAs no tienen implementados circuitos comunes de la lista")
                        End If 'No hubo circuitos comunes

                        'Valores comunes a ambas situaciones para mostrar en pantalla
                        texto_para_text2 = "Todas las FPGAs con circuitos comunes implementados"
                        texto_para_text1 = "Global"
                        variable_de_circuitos = circuitos_comunes_a_todas

                        'c) Todas las FPGAs filtradas con los circuitos seleccionados'
                        '-----------------------------------------------------------
                        ElseIf Form1.ComboBox12.SelectedIndex = 2 Then

                            'Se determinan los circuitos seleccionados para trabajar con ellos
                            '-------------------------------------------------------------------
                            Dim array_circuitos_seleccionados() As String = {}

                            If Form1.TextBox1.Text <> "" Then
                                array_circuitos_seleccionados = Form1.TextBox1.Text.Split(",")
                            Else
                                MsgBox("No ha seleccionado ningún circuito", , "Información")
                                no_seleccion = True
                            Form1.Show()
                            End If

              'Se comprueba si los circuitos seleccionados se han implmentado en todas las tecnologias
              'seleccionadas, para quedarnos solo con aquellos que lo hayan conseguido
              '-----------------------------------------------------------------------------------------
                        Dim circuitos_comunes_de_seleccionados() As String = {}
                        Dim tec_con_todos_circ() As String = {}
                        ReDim Preserve matriz_tec_punt(7, array_fpga_criterios.Count - 1)
          circuitos_comunes_de_seleccionados = todas_tecnologias_circ_comunes(array_fpga_criterios, _
array_circuitos_seleccionados)
          tec_con_todos_circ = comprueba_tecnologia_circ_imp(array_fpga_criterios, _
array_circuitos_seleccionados)

                        If circuitos_comunes_de_seleccionados.Count <> 0 Then
                            'Si existec circuitos comunes atodas las FPGAs se sigue con ello
     matriz_tec_punt = Punt_lista_tec_circ(array_fpga_criterios, circuitos_comunes_de_seleccionados)
```

```
                        'Muestra los resultados
                        '-----------------------
                        array_indice = array_fpga_criterios
                        variable_de_circuitos = circuitos_comunes_de_seleccionados

                    ElseIf tec_con_todos_circ.Count <> 0 Then
                        'Si no, se comprueba qué tecnologías de la lista q  aqui llega
                        'tienen los circuito seleccionados en comun
        matriz_tec_punt = Punt_lista_tec_circ(tec_con_todos_circ, array_circuitos_seleccionados)
                        array_indice = tec_con_todos_circ
                        variable_de_circuitos = array_circuitos_seleccionados
        MsgBox("De las FPGAs elegidas, se muestran solo las que implementaron todos los circuitos")
                    End If

                    'Valores comunes a ambas situaciones para mostrar en pantalla
                    texto_para_text2 = "FPGAs con los circuitos seleccionados implementados"
                    texto_para_text1 = "Seleccionados"
            End If '--Fin de combinaciones con GLOBAL

            'SI SE SELECCIONA ALGUN SECTOR
            '-----------------------------------------------------------------------------
        Else
                'se cargan las tablas que incluyen el listado de circuitos de cada uno
Bench_autoTableAdapter.Fill(Resultados_benchamrkDataSet.bench_auto)
Bench_consumoTableAdapter.Fill(Resultados_benchamrkDataSet.bench_consumo)
Bench_comunicacionesTableAdapter.Fill(Resultados_benchamrkDataSet.bench_comunicaciones)

'Mediante la función correspondiente se almacena en sendos array la lista de circuitos de cada sector
-------------------------------------------------------------------------------------------------------
                Dim array_auto() As String = {}
                Dim array_consumo() As String = {}
                Dim array_comunicaciones() As String = {}

                array_auto = array_desde_tabla(Bench_autoBindingSource)
                array_consumo = array_desde_tabla(Bench_consumoBindingSource)
                array_comunicaciones = array_desde_tabla(Bench_comunicacionesBindingSource)

                'En función del sector seleccionado se tomará un camino diferente
                '----------------------------------------------------------------

                Select Case Form1.ComboBox6.SelectedIndex
                    '''''''''''''
                    'Automocion'
                    '''''''''''''

                    Case 1
                        'Comprueba si la tabla de AUTO contiene circuitos

                        If Bench_autoBindingSource.Count <> 0 Then

                            'a) Todos los circuitos
                            '-------------------
                            If Form1.ComboBox12.SelectedIndex = 0 Then
                                Dim tec_todos_circuitos() As String
            tec_todos_circuitos = comprueba_tecnologia_circ_imp(array_fpga_criterios, array_auto)

             'si existen FPGAs con todos los circuitos implementados se calculan las puntuaciones
                                If tec_todos_circuitos.Count <> 0 Then
                                 ReDim Preserve matriz_tec_punt(7, tec_todos_circuitos.Count - 1)
                        matriz_tec_punt = Punt_lista_tec_circ(tec_todos_circuitos, array_auto)

                                    'Muestra los resultados
                                    '-----------------------
                                    array_indice = tec_todos_circuitos
                                Else
   MsgBox("Ninguna tecnología de las seleccionadas ha implementado todos los circuitos del sector")
                                End If

                                'Valores comunes a ambas situaciones para mostrar en pantalla
                                texto_para_text2 = "FPGAs con todos los circuitos implementados"
                                texto_para_text1 = "Automoción"
                                variable_de_circuitos = array_auto


                                'b) todas las FPGAs
                                '-----------------
                            ElseIf Form1.ComboBox12.SelectedIndex = 1 Then
                                Dim circuitos_comunes_a_todas() As String = {}
        circuitos_comunes_a_todas = todas_tecnologias_circ_comunes(array_fpga_criterios, array_auto)
                                'Si existen circuitos comunes calcula las puntuaciones

                                ReDim Preserve matriz_tec_punt(7, array_fpga_criterios.Count - 1)
                                If circuitos_comunes_a_todas.Count <> 0 Then
        matriz_tec_punt = Punt_lista_tec_circ(array_fpga_criterios, circuitos_comunes_a_todas)

                                    'Muestra resultados
```

```
                                    '-------------------------------------------------------------
                                    array_indice = array_fpga_criterios
                                Else
MsgBox("Ninguna tecnología de las seleccionadas ha implementado circuitos comunes del sector")
                                End If

                                'Valores comunes a ambas situaciones para mostrar en pantalla
                        texto_para_text2 = "Todas las FPGAs con circuitos comunes implementados"
                        texto_para_text1 = "Automoción"
                        variable_de_circuitos = circuitos_comunes_a_todas
                                End If

                            Else
                                MsgBox("Elija en ResultadosPdC circuitos para este sector")
                            End If '--Comprobacion de q la tabla bench_auto tiene circuitos

                            '''''''''''''
                            'Consumo    '
                            '''''''''''''
                        Case 2
                            'a) Todos los circuitos
                            '--------------------
                            If Form1.ComboBox12.SelectedIndex = 0 Then
                                Dim tec_todos_circuitos() As String
          tec_todos_circuitos = comprueba_tecnologia_circ_imp(array_fpga_criterios, array_consumo)

           'si existen FPGAs con todos los circuitos implementados se calculan las puntuaciones
                                If tec_todos_circuitos.Count <> 0 Then
                                    ReDim Preserve matriz_tec_punt(7, tec_todos_circuitos.Count - 1)
                         matriz_tec_punt = Punt_lista_tec_circ(tec_todos_circuitos, array_consumo)

                                    'Muestra los resultados
                                    '-----------------------
                                    array_indice = tec_todos_circuitos
                                Else
  MsgBox("Ninguna tecnología de las seleccionadas ha implementado todos los circuitos del sector")
                                End If

                                texto_para_text2 = "FPGAs con todos los circuitos implementados"
                                texto_para_text1 = "Consumo"
                                variable_de_circuitos = array_consumo


                                'b) todas las FPGAs
                                '-----------------
                            ElseIf Form1.ComboBox12.SelectedIndex = 1 Then
                                Dim circuitos_comunes_a_todas() As String = {}
                                ReDim Preserve matriz_tec_punt(7, array_fpga_criterios.Count - 1)
 circuitos_comunes_a_todas = todas_tecnologias_circ_comunes(array_fpga_criterios, array_consumo)

                                'Si existen circuitos comunes calcula las puntuaciones
                                  If circuitos_comunes_a_todas.Count <> 0 Then
         matriz_tec_punt = Punt_lista_tec_circ(array_fpga_criterios, circuitos_comunes_a_todas)

                                    'Muestra resultados
                                    '-------------------------------------------------------------
                                    array_indice = array_fpga_criterios
                                  Else
   MsgBox("Ninguna tecnología de las seleccionadas ha implementado circuitos comunes del sector")
                                  End If

                                texto_para_text2 = "FPGAs con todos los circuitos implementados"
                                texto_para_text1 = "Consumo"
                                variable_de_circuitos = circuitos_comunes_a_todas

                            End If '--Fin Consumo


                            '''''''''''''''''''
                            'Comunicaciones    '
                            '''''''''''''''''''
                        Case 3
                            'a) Todos los circuitos
                            '--------------------
                            If Form1.ComboBox12.SelectedIndex = 0 Then
                                Dim tec_todos_circuitos() As String
   tec_todos_circuitos = comprueba_tecnologia_circ_imp(array_fpga_criterios, array_comunicaciones)

           'si existen FPGAs con todos los circuitos implementados se calculan las puntuaciones
                                If tec_todos_circuitos.Count <> 0 Then
                                    ReDim Preserve matriz_tec_punt(7, tec_todos_circuitos.Count - 1)
               matriz_tec_punt = Punt_lista_tec_circ(tec_todos_circuitos, array_comunicaciones)

                                    'Muestra los resultados
                                    '-----------------------
                                    array_indice = tec_todos_circuitos
```

```
                               Else
    MsgBox("Ninguna tecnología de las seleccionadas ha implementado todos los circuitos del sector")
                                End If

                                texto_para_text2 = "FPGAs con todos los circuitos implementados"
                                texto_para_text1 = "Comunicaciones"
                                variable_de_circuitos = array_comunicaciones


                                'b) todas las FPGAs
                                '-----------------
                            ElseIf Form1.ComboBox12.SelectedIndex = 1 Then
                                Dim circuitos_comunes_a_todas() As String = {}
                                ReDim Preserve matriz_tec_punt(7, array_fpga_criterios.Count - 1)

        circuitos_comunes_a_todas = todas_tecnologias_circ_comunes(array_fpga_criterios,
array_comunicaciones)
                                'Si existen circuitos comunes calcula las puntuaciones
                                If circuitos_comunes_a_todas.Count <> 0 Then
             matriz_tec_punt = Punt_lista_tec_circ(array_fpga_criterios, circuitos_comunes_a_todas)

                                    'Muestra resultados
                                    '-------------------------------------------------------------
                                    array_indice = array_fpga_criterios
                                Else
      MsgBox("Ninguna tecnología de las seleccionadas ha implementado circuitos comunes del sector")
                                End If

                                texto_para_text2 = "FPGAs con todos los circuitos implementados"
                                texto_para_text1 = "Comunicaciones"
                                variable_de_circuitos = circuitos_comunes_a_todas

                            End If '--Fin comunicaciones
                    End Select '--Fin Sector seleccionado
                End If

                    If (no_seleccion = False And array_indice.Count <> 0) Then
                        Dim matriz_caracteristicas(6, array_indice.Count - 1) As String
                        matriz_caracteristicas = caracteristicas_tec_filtradas(array_indice)

                        'Muestra resultados
                        '-------------------------------------------------------------
                        Dim total_registros_matriz_tec_punt As Integer
                        total_registros_matriz_tec_punt = matriz_tec_punt.GetLength(1)
                        Dim caracteristicas_fpga(,) As String
                        Dim array_tec(0) As String

                        For x = 0 To total_registros_matriz_tec_punt - 1
                            tabla_FPGAs.tecnologia = matriz_tec_punt(0, x)
                            tabla_FPGAs.circuito = matriz_tec_punt(1, x)
                            tabla_FPGAs.P1_global = CSng(matriz_tec_punt(2, x))
                            tabla_FPGAs.P2_global = CSng(matriz_tec_punt(3, x))
                            tabla_FPGAs.P3_global = CSng(matriz_tec_punt(4, x))
                            tabla_FPGAs.P4_global = CSng(matriz_tec_punt(5, x))
                            tabla_FPGAs.P5_global = CSng(matriz_tec_punt(6, x))
                            tabla_FPGAs.P_total = CSng(matriz_tec_punt(7, x))

                            'se determina el precio de cada FPGA
                            array_tec(0) = tabla_FPGAs.tecnologia
                            caracteristicas_fpga = caracteristicas_tec_filtradas(array_tec)
                            tabla_FPGAs.precio = CSng(caracteristicas_fpga(1, 0))

          Tabla_seleccion_FPGATableAdapter.Insert(tabla_FPGAs.tecnologia, tabla_FPGAs.circuito,
tabla_FPGAs.P1_global, tabla_FPGAs.P2_global, tabla_FPGAs.P3_global, tabla_FPGAs.P4_global,
tabla_FPGAs.P5_global, tabla_FPGAs.P_total, tabla_FPGAs.precio)
                        Next
                    End If

                Else '--si ninguna FPGA se ajusta a los criterios elegidos
                    MsgBox("Ninguna FPGA cumple esos requisitos")
                End If
        End If '--Bandera viene_de_volver

        'Muestra los valores almacenado en Tabla_seleccion_FPGA
        'tanto la primera vez, como si viene tras pulsar VOLVER o NUEVA BUSQUEDA
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        Tabla_seleccion_FPGATableAdapter.Fill(Resultados_benchamrkDataSet.Tabla_seleccion_FPGA)
        TextBox3.Text = Tabla_seleccion_FPGABindingSource.Count
        TextBox2.Text = texto_para_text2
        TextBox1.Text = texto_para_text1
        TextBox4.Text = variable_de_circuitos.Count
        For k = 0 To variable_de_circuitos.Count - 1
            ListView2.Items.Add(variable_de_circuitos(k))
        Next
```

```
    End Sub

Private Function comprueba_tecnologia_circ_imp(ByVal lista_tecnologias() As String, ByVal
lista_circuitos() As String)
'Esta función recibe una lista de tecnologias y comprueba cuales de ellas ha podido implementar todos
' los circuitos de Lista_circuitos
----------------------------------------------------------------------------------------------------
Me.Circuitos_implementadosTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Circuitos_implementados)

        Dim total As Integer = 0
        Dim fila_circuitos_implementados As DataRowView
        Dim tecnologia As String
        Dim total_circuitos_implementados As Integer
        Dim array_circuitos_implementados() As String
        Dim circuitos_implementados As String
        Dim cont_tec As Integer = 0
        Dim tecnologias_lista_circuitos() As String = {}

        total_circuitos_implementados = Circuitos_implementadosBindingSource.Count

        'Por cada tecnologia de Lista_Tecnologias se accede a la tabla CIRCUITOS_IMPLEMENTADOS
        'para comprobar si sobre ella se han implementado todos los circuitos de Lista_circuitos
        '----------------------------------------------------------------------------------------
        Dim paquito As Integer = 0
        For i = 0 To lista_tecnologias.Count - 1
            Dim aviso_encontrada As Boolean = False
            total = 0
     'Comprueba Si la tecnologia i coincide con alguna de CIRCUITOS_IMPLEMENTADOS
     '------------------------------------------------------------------------------------------------
            While ((total < total_circuitos_implementados - 1) And (aviso_encontrada = False))
                fila_circuitos_implementados = Circuitos_implementadosBindingSource.Item(total)
                tecnologia = fila_circuitos_implementados.Row.ItemArray(1)
                circuitos_implementados = fila_circuitos_implementados.Row.ItemArray(2)

                If lista_tecnologias(i) = tecnologia Then
                    aviso_encontrada = True

                    'Comprobación de que array_circuitos_implementados contiene a "lista_circuitos"
                    '-------------------------------------------------------------------------------
                    Dim circuitos_comunes As Integer = 0

                    array_circuitos_implementados = circuitos_implementados.Split(",")
                    For j = 0 To lista_circuitos.Count - 1
                        If array_circuitos_implementados.Contains(lista_circuitos(j)) Then
                            circuitos_comunes = circuitos_comunes + 1
                        End If
                    Next

                    If circuitos_comunes = lista_circuitos.Count Then
                        ReDim Preserve tecnologias_lista_circuitos(cont_tec)
                        tecnologias_lista_circuitos(cont_tec) = tecnologia
                        cont_tec = cont_tec + 1
                    End If
                    total = total + 1
                Else
                    paquito = paquito + 1
                    total = total + 1
                End If
            End While
        Next
        Return tecnologias_lista_circuitos
    End Function

Private Function array_desde_tabla(ByVal tabla As BindingSource)
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        'Genera un array a partir de los elementos de la tabla seleccionada
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        Dim fila_sector As DataRowView
        Dim total_circ As Integer
        'total de circuitos de componen el sector elegido
        total_circ = tabla.Count

        Dim array_circuitos(total_circ - 1) As String

        For i = 0 To total_circ - 1
            fila_sector = tabla.Item(i)
            array_circuitos(i) = fila_sector.Row.ItemArray(1)
            array_circuitos(i) = array_circuitos(i).ToUpper
        Next

        Return array_circuitos
    End Function

Private Function Punt_lista_tec_circ(ByVal lista_tecnologias() As String, ByVal lista_Circuitos() As
String)
        ' Recibe una lista de tecnologias y circuitos para los que se quiere obtener una   '
```

```
        ' puntuación leyendo la tabla NORMALIZADOS. Devuelve   una matriz que contiene     '
        ' por cada tecnologia de lista_tecnologias sus cinco puntuaciones, la puntuación   '
        ' global calculada a partir de las anteriores y la lista_circuitos.
        '-----------------------------------------------------------------------------------

        Dim total_normalizados As Integer
        Dim fila_normalizados As DataRowView
        Dim total_lista_tecnologias As Integer
        Dim tecnologia As String
        Dim estrategia As String
        Dim circuito As String
        Dim matriz_sector(7, 0) As String
        Dim indice_matriz As Integer = 0
        Dim array_Pk(4) As Single
        Dim PG As Single

        NormalizadosTableAdapter.Fill(Resultados_benchamrkDataSet.Normalizados)
        total_normalizados = NormalizadosBindingSource.Count
        total_lista_tecnologias = lista_tecnologias.Count

        'Por cada tecnología de lista_tecnologia, se localiza en NORMALIZADOS y en el caso de que
        'el campo "estrategia"=balanced y el campo "circuito" coincida con alguno de lista_circuitos
        'Se almacenan las 5 puntuaciones en su correspondiente array
        '-------------------------------------------------------------------------------------------
        Dim h As Integer = 0
        For i = 0 To total_lista_tecnologias - 1
            'Se avanza por cada una de las tecnologias(registros) que componen NORMALIZADOS
       ´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´

            'Se busca la tecnologia en NORMALIZADOS y si coincide tambien la estrategia con BALANCED
            'se comprueba si el circuito de ese registro pertenece a lista_circuito
            '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´

            For j = 0 To total_normalizados - 1
                fila_normalizados = NormalizadosBindingSource.Item(j)
                estrategia = fila_normalizados.Row.ItemArray(2)
                estrategia = estrategia.ToUpper
                circuito = fila_normalizados.Row.ItemArray(1)
                circuito = circuito.ToUpper
                tecnologia = fila_normalizados.Row.ItemArray(3)

            If (fila_normalizados.Row.ItemArray(3) = lista_tecnologias(i) And estrategia = "BALANCED"
And lista_Circuitos.Contains(circuito)) Then

                    'SI:se calcula la puntuación global PG para esa tecnología y circuito
                    '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´

                    array_Pk(0) = fila_normalizados.Row.ItemArray(4)
                    array_Pk(1) = fila_normalizados.Row.ItemArray(5)
                    array_Pk(2) = fila_normalizados.Row.ItemArray(6)
                    array_Pk(3) = fila_normalizados.Row.ItemArray(7)
                    array_Pk(4) = fila_normalizados.Row.ItemArray(8)

                    PG = media_geometrica(array_Pk)

                    ReDim Preserve matriz_sector(7, h)

                    matriz_sector(0, h) = tecnologia
                    matriz_sector(1, h) = circuito
                    matriz_sector(2, h) = CStr(array_Pk(0))
                    matriz_sector(3, h) = CStr(array_Pk(1))
                    matriz_sector(4, h) = CStr(array_Pk(2))
                    matriz_sector(5, h) = CStr(array_Pk(3))
                    matriz_sector(6, h) = CStr(array_Pk(4))
                    matriz_sector(7, h) = CStr(PG)

                    h = h + 1
                End If
            Next
        Next
        Return matriz_sector
    End Function

Public Function media_geometrica(ByVal array_datos() As Single)
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ' Calcula la media geometrica de un array de valores doble precisión '
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        Dim n As Integer
        Dim raiz As Single
        Dim producto As Single
        Dim media As Single
        Dim total_array_datos As Integer

        'Si el array recibido no está vacio realiza la media geometrica,
        'en caso cotnrario devuelve el valor 0

        If array_datos Is Nothing Then
            media = 0
```

```
        Else
            'Obtiene la longitud del array pasado para saber el total de elemntos
            'para el que se calculará la media geometrica
            '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            total_array_datos = array_datos.Length
            n = array_datos.Length
            raiz = 1 / n

            'Calcula el producto de los n elementos
            '''''''''''''''''''''''''''''''''''''''
            producto = 1
            For i = 0 To n - 1
                producto = producto * array_datos(i)
            Next

            'Obtiene la media geometrica
            ''''''''''''''''''''''''''''
            media = producto ^ raiz
        End If
        Return media
    End Function

 Private Function todas_tecnologias_circ_comunes(ByVal lista_tecnologias() As String, _
                                                    ByVal lista_circuitos() As String)
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ' Devuelve un array con los circuitos que se hayan implementado en todas las
        ' tecnologias de la lista que se le indique. Los circuitos de referencia son   '
        ' lista_circuitos
      '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        Circuitos_implementadosTableAdapter.Fill(Resultados_benchamrkDataSet.Circuitos_implementados)

        Dim total_lista_circuitos As Integer
        Dim total_circuitos_implementados As Integer
        Dim fila_circuitos_implementados As DataRowView
        Dim total_lista_fpgas As Integer
        Dim cadena_circuitos As String
        Dim contador As Integer = 0
        Dim array_circuitos_comunes() As String = {}
        Dim k As Integer = 0

        total_lista_circuitos = lista_circuitos.Count
        total_circuitos_implementados = Circuitos_implementadosBindingSource.Count
        total_lista_fpgas = lista_tecnologias.Count

        'Se comprueba si cada uno de los circuitos de lista_circuitos
        'se ha implementado en todas las FPGAs (Tabla:CIRCUITOS_IMPLEMENTADOS)
        '´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´´
        For i = 0 To total_lista_circuitos - 1
            contador = 0
            For j = 0 To total_circuitos_implementados - 1 'CIRCUITOS_IMPLEMENTADOS
                fila_circuitos_implementados = Circuitos_implementadosBindingSource.Item(j)
                'Si la tecnologia coincide con alguna de la lista
                If lista_tecnologias.Contains(fila_circuitos_implementados.Row.ItemArray(1)) Then
                    'ESta variable guarda la cadena de circuitos implementados en cada tecnologia
                    cadena_circuitos = fila_circuitos_implementados.Row.ItemArray(2)
                    cadena_circuitos = cadena_circuitos.ToUpper

                    'Si el circuito i está en esa cadena se incrementa un contador
                    lista_circuitos(i) = lista_circuitos(i).Trim
                    If cadena_circuitos.Contains(lista_circuitos(i)) Then
                        contador = contador + 1
                    End If
                End If
            Next

            'Si el contador coincide con el total de tecnologías se guarda el circuito
            'en un array, denotando que ese circuito se ha implementado en todas las tecnologias
            '-----------------------------------------------------------------------------------
            If contador = total_lista_fpgas Then
                ReDim Preserve array_circuitos_comunes(k)
                array_circuitos_comunes(k) = lista_circuitos(i)
                array_circuitos_comunes(k) = array_circuitos_comunes(k).ToUpper
                k = k + 1
            End If
        Next
        Return array_circuitos_comunes
    End Function

Private Function caracteristicas_tec_filtradas(ByVal lista_tecnologias() As String)
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ' Devuelve una matriz con todas las características principales de las tecnologias
        ' de lista_tecnologias
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        Dim matriz_tec_caracteristicas(8, 0) As String
        Dim total As Integer
        Dim fila_tabla As DataRowView
```

```
        Dim total_tabla As Integer
        Dim aviso_encontrado As Boolean = False
        Dim array_caracteristicas(6) As String
        Dim total_matriz As Integer = 0

        total_tabla = CaracteristicasBindingSource.Count

        For i = 0 To lista_tecnologias.Count - 1
            total = 0
            aviso_encontrado = False
            While (total < total_tabla And aviso_encontrado = False)
                fila_tabla = CaracteristicasBindingSource.Item(total)
                'cuando coincida la tecnologia con la del registro actual de la tabla
                'se guardan todas sus caracteristicas
                If lista_tecnologias(i) = fila_tabla.Row.ItemArray(1) Then
                    ReDim Preserve matriz_tec_caracteristicas(8, total_matriz)
             matriz_tec_caracteristicas(0, total_matriz) = fila_tabla.Row.ItemArray(1) '--modelo
             matriz_tec_caracteristicas(1, total_matriz) = fila_tabla.Row.ItemArray(2) '--precio
             matriz_tec_caracteristicas(2, total_matriz) = fila_tabla.Row.ItemArray(3) '--encapsulado
             matriz_tec_caracteristicas(3, total_matriz) = fila_tabla.Row.ItemArray(5) '--UL
             matriz_tec_caracteristicas(4, total_matriz) = fila_tabla.Row.ItemArray(8) '--bits
             matriz_tec_caracteristicas(5, total_matriz) = fila_tabla.Row.ItemArray(9) '--bits
             matriz_tec_caracteristicas(6, total_matriz) = fila_tabla.Row.ItemArray(11) '--MULT
             matriz_tec_caracteristicas(7, total_matriz) = fila_tabla.Row.ItemArray(13) '--Tmax
             matriz_tec_caracteristicas(8, total_matriz) = fila_tabla.Row.ItemArray(14) '--Tmin

                    total_matriz = total_matriz + 1
                    aviso_encontrado = True
                    total = total + 1
                Else
                    total = total + 1
                End If
            End While
        Next
        Return matriz_tec_caracteristicas
    End Function

Private Function conjunto_valor_columna(ByVal tabla As BindingSource, ByVal columna As Integer)
'Devuelve un array que contiene los elementos, sin repetir, de la columna 1 de la tabla especificada
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        Dim fila As DataRowView
        Dim total_tabla As Integer
        Dim array_almacen() As String = {}
        Dim valor_columna As String
        Dim contador_valor_columna As Integer = 0

        total_tabla = tabla.Count

        For i = 0 To total_tabla - 1
            'Se carga una fila de la tabla indicada
            fila = tabla.Item(i)

            'Conversion a mayúsculas
            valor_columna = fila.Row.ItemArray(columna)
            valor_columna = valor_columna.ToUpper

            'Si el valor_columna del registro actual no se ha guardado todavia en array_almacen
            'se guarda
            If Not array_almacen.Contains(valor_columna) Then
                ReDim Preserve array_almacen(contador_valor_columna)
                array_almacen(contador_valor_columna) = valor_columna
                contador_valor_columna = contador_valor_columna + 1
            End If
        Next
        Return array_almacen
    End Function
Private Function array_por_registro(ByVal tabla As BindingSource, ByVal col_busqueda As Integer,
ByVal criterio As String, ByVal col1 As Integer)
     ' Devuelve un array (single) que contiene todos los valores de "col1" correspondientes a todos l
     ' los(registros)cuyo campo "col_busqueda" coincida con "criterio", a partir de una tabla "tabla"
    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        Dim fila As DataRowView
        Dim total_tabla As Integer
        Dim array_campo() As Single = {}
        Dim cont_valor As Integer = 0

        total_tabla = tabla.Count
        criterio = criterio.ToUpper

        For i = 0 To total_tabla - 1
            'Se carga una fila de la tabla indicada
            fila = tabla.Item(i)
            If fila.Row.ItemArray(col_busqueda) = criterio Then
                ReDim Preserve array_campo(cont_valor)
```

```
                array_campo(cont_valor) = fila.Row.ItemArray(col1)
                cont_valor = cont_valor + 1
            End If
        Next
        Return array_campo
    End Function

Private Sub ToolStripButton3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ToolStripButton3.Click
        '''''''''''''''''''''''
        ' PUNTUACION FINAL ''''
        '''''''''''''''''''''''
        Tabla_seleccion_FPGA_PFTableAdapter.DeleteQuery()

        Dim array_tec_tabla() As String = {}
        Dim array_aux_pg() As Single
        Dim PF As Single
        Dim carac_fpga(,) As String = {}
        Dim array_temp(0) As String

        'Obtiene la lista de tecnologias que interesa buscar en la tabla
        'utilzando la funcion que devuelve, sin repetir, todos los valores
        'del campo 1 de la tabla tabla
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        array_tec_tabla = conjunto_valor_columna(Tabla_seleccion_FPGABindingSource, 1)


        'Busca las tecnologias en la tabla de nuevo, una por una "anotando" el valor de PG que
        'le corresponde ante cada circuito. Luego obtiene la media geometrica de ellos (PF)
        'Repite la operación de búsqueda para obtener el precio, aunque es innecesario hallar
        'media geometrica, se tiene como el proceso más simple en este punto.
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        For i = 0 To array_tec_tabla.Count - 1
       array_aux_pg = array_por_registro(Tabla_seleccion_FPGABindingSource, 1, array_tec_tabla(i), 8)
            PF = media_geometrica(array_aux_pg)
            array_temp(0) = array_tec_tabla(i)
            carac_fpga = caracteristicas_tec_filtradas(array_temp)
                  Tabla_seleccion_FPGA_PFTableAdapter.Insert(array_tec_tabla(i), "0", PF,
                 carac_fpga(1, 0),
                 carac_fpga(4, 0), carac_fpga(3, 0), carac_fpga(5, 0),
                 carac_fpga(6, 0), carac_fpga(2, 0), carac_fpga(7, 0), carac_fpga(8, 0))
        Next
        Form4.Show()
        Me.Close()
    End Sub

End Class
```

Form3.VB

```
Public Class Principal
    Public pines_max As Integer
    Public tem_max_max As Integer
    Public temp_min_min As Integer
    Public ul_max As Integer
    Public precio_max As Single

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        Button1.Enabled = False
        'Borrado inicial de todas las tablas donde se insertarán datos
        Me.PrecioTableAdapter.DeleteQuery()
        Me.EncapsuladosTableAdapter.DeleteQuery()
        Me.PinesTableAdapter.DeleteQuery()
        Me.Unidades_logicasTableAdapter.DeleteQuery()
        Me.Temp_minTableAdapter.DeleteQuery()
        Me.Temp_maxTableAdapter.DeleteQuery()

        'Carga de la tabla de la que se usarán datos
        Me.CaracteristicasTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Caracteristicas)

        'Llamada a la función datos_tabla por cada caracteristica que se ofrecera como
        'filtro de búsqueda

        ''''''''
        ' PINES'
        ''''''''
        'Se trabaja con enteros que es tal y como estan almacenados en la tabla de la BD
        Dim array_pines() As Integer
        array_pines = datos_tabla(CaracteristicasBindingSource, 8, "entero")
```

```
        'Almacena el maximo valor encontrado para usarlo posteriormente
        pines_max = array_pines.Min

        'Inserta un valor en blanco como elemento por defecto para que luego aparezca en el combobox
        PinesTableAdapter.Insert("")

        'Inserta todos los elementos del array en la tabla correspondiente, fuente para el combobox
        For i = 0 To array_pines.Length - 1
            PinesTableAdapter.Insert(array_pines(i))
        Next


        ''''''''
        ' T_MAX'
        ''''''''
        Dim array_temperatura_max() As Integer
        array_temperatura_max = datos_tabla(CaracteristicasBindingSource, 13, "entero")
        tem_max_max = array_temperatura_max.Min

        Temp_maxTableAdapter.Insert("")
        For i = 0 To array_temperatura_max.Length - 1
            Temp_maxTableAdapter.Insert(array_temperatura_max(i))
        Next

        ''''''''
        ' T_MIN'
        ''''''''
        Dim array_temperatura_min() As Integer

        array_temperatura_min = datos_tabla(CaracteristicasBindingSource, 14, "entero")
        temp_min_min = array_temperatura_min.Max
        Temp_minTableAdapter.Insert("")
        For i = 0 To array_temperatura_min.Length - 1
            Temp_minTableAdapter.Insert(array_temperatura_min(i))
        Next

        ''''''''''''''
        ' ENCAPSULADO'
        ''''''''''''''
        Dim array_encapsulado() As String
        array_encapsulado = datos_tabla(CaracteristicasBindingSource, 3, "string")
        EncapsuladosTableAdapter.Insert("")
        For i = 0 To array_encapsulado.Length - 1
            EncapsuladosTableAdapter.Insert(array_encapsulado(i))
        Next

        ''''''''
        ' UL   '
        ''''''''
        Dim array_UL() As Integer
        array_UL = datos_tabla(CaracteristicasBindingSource, 5, "entero")
        ul_max = array_UL.Min
        Unidades_logicasTableAdapter.Insert("")
        For i = 0 To array_UL.Length - 1
            Unidades_logicasTableAdapter.Insert(array_UL(i))
        Next

        ''''''''
        'PRECIO'
        ''''''''
        Dim array_precio() As Single
        array_precio = datos_tabla(CaracteristicasBindingSource, 2, "single")
        Array.Reverse(array_precio)
        precio_max = array_precio.Max
        'PrecioTableAdapter.Insert(0)
        For i = 0 To array_precio.Length - 1
            PrecioTableAdapter.Insert(array_precio(i))
        Next

        MsgBox("La operación ha terminado")
        Form1.Show()
        Me.Hide()
    End Sub

Private Function datos_tabla(ByVal tabla As BindingSource, ByVal numero_campo As Integer, ByVal
tipo_dato As String)
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ' Devuelve en forma de array ordenado los datos de una columna de una tabla '
        ' Sin repetir ningun valor. Parámetros: tabla, numero de columna,tipode dato'
        ' Si se trata de enteros toma un camino y otro si es string esto se debe a que si
        ' queremos ordenar valores numéricos no pueden estar definidos  como String.
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        'Variable que apunta a un registro de una tabla
        Dim fila_tabla As DataRowView
        Dim total_tabla As Integer
```

```
        'Este array guardará todos los campo disponibles para ls FPGAs tal y como aparecen en la
tabla CARACTERISTICAs
        'incluso si están repetidos
        Dim campo_array_int() As Integer = {}
        Dim campo_array_stg() As String = {}
        Dim campo_array_sng() As Single = {}
        Dim contador_campo As Integer = 0
        Dim campo As String

        total_tabla = CaracteristicasBindingSource.Count

        For i = 0 To total_tabla - 1
            fila_tabla = tabla.Item(i)
            campo = fila_tabla.Row.ItemArray(numero_campo)
            tipo_dato = tipo_dato.ToUpper
            If tipo_dato = "ENTERO" Then
                campo = CInt(campo)
                'si el valor no ha sido aún almacenado se guarda en campo_array
                If Not campo_array_int.Contains(campo) Then
                    ReDim Preserve campo_array_int(contador_campo)
                    campo_array_int(contador_campo) = campo
                    contador_campo = contador_campo + 1
                End If
            ElseIf tipo_dato = "STRING" Then
                If Not campo_array_stg.Contains(campo) Then
                    ReDim Preserve campo_array_stg(contador_campo)
                    campo_array_stg(contador_campo) = campo
                    contador_campo = contador_campo + 1
                End If
            ElseIf tipo_dato = "SINGLE" Then
                campo = CSng(campo)
                If Not campo_array_sng.Contains(campo) Then
                    ReDim Preserve campo_array_sng(contador_campo)
                    campo_array_sng(contador_campo) = campo
                    contador_campo = contador_campo + 1
                End If
            End If
        Next

        If tipo_dato = "ENTERO" Then
            Array.Sort(campo_array_int)
            Return campo_array_int
        ElseIf tipo_dato = "STRING" Then
            Array.Sort(campo_array_stg)
            Return campo_array_stg
        ElseIf tipo_dato = "SINGLE" Then
            Array.Sort(campo_array_sng)
            Return campo_array_sng
        End If
        Return 0
End Function

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
        Application.Exit()
End Sub
End Class
```

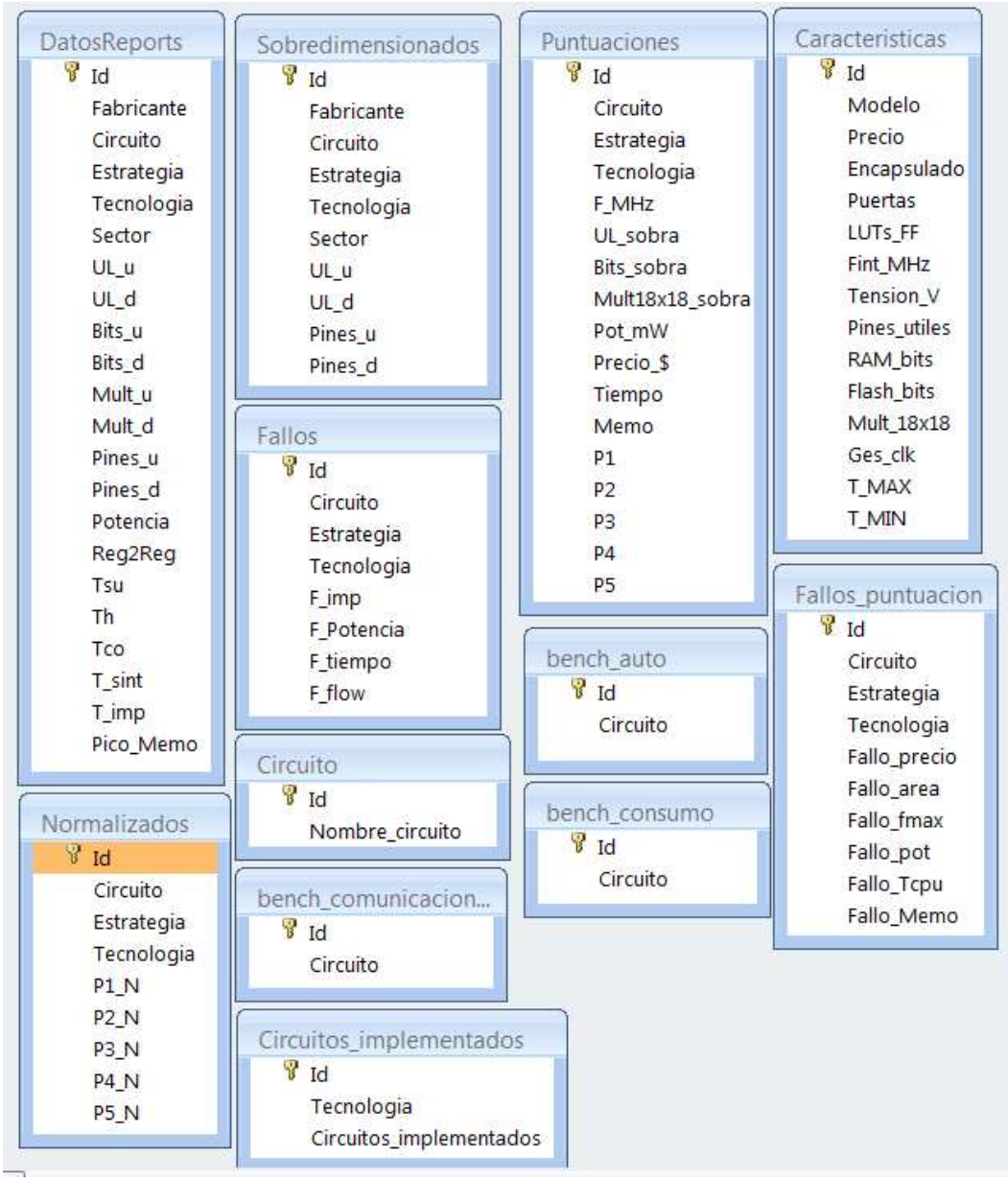
Form4.vb

```
Public Class Form4

Private Sub Form4_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load

Me.Tabla_seleccion_FPGA_PFTableAdapter.Fill(Me.Resultados_benchamrkDataSet.Tabla_seleccion_FPGA_PF)
        TextBox1.Text = Tabla_seleccion_FPGA_PFBindingSource.Count
End Sub

Private Sub VolverToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles VolverToolStripMenuItem.Click
        viene_de_volver = True
        Form2.Show()
        Me.Close()
End Sub

Private Sub SalirToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles SalirToolStripMenuItem.Click
        Application.Exit()
End Sub

End Class
```

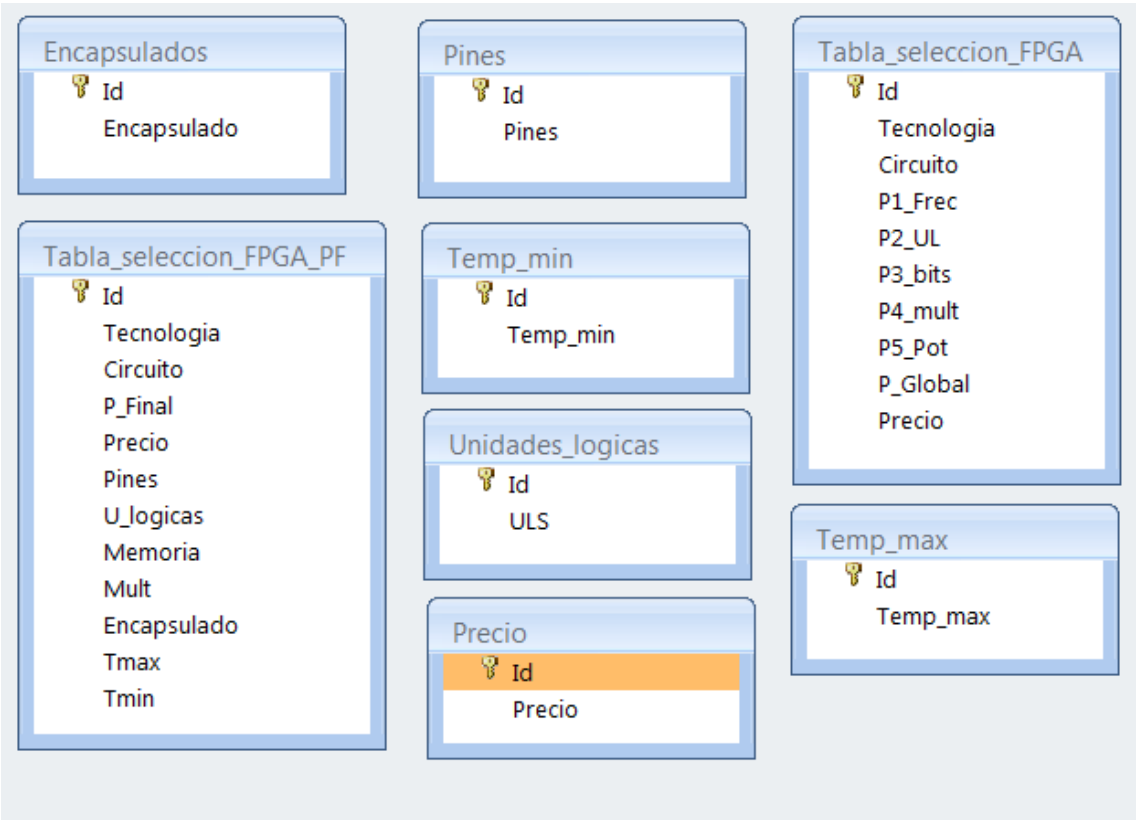
Anexo C: Tablas de la Base de Datos

En este apartado se adjunta el total de tablas contenidos en la base de datos, sin contenido por razones de espacio. Se ofrece para disponer de una guía rápida de los campos que las componen.

ResultadosPDC



AsistenteFPGA



Anexo D: Manual de usuario de las aplicaciones

Instalación

Los requisitos de la instalación se recogen en la Tabla 109.

Sistema Operativo	Windows XP (SP2), Windows Vista
Microsoft .NET Framework	3.0
Microsoft Access	2007

Tabla 109. Requisitos previos de la instalación

El fichero de instalación se denomina Instalación_PdC_01.msi y es un paquete de Windows Installer que incluye las aplicaciones **ResultadosPdC y AsistenteFPGA**. La ejecución del fichero lanza un asistente que guiará el proceso de instalación, Figura 72.

Figura 72. Imagen del asistente de instalación de PdC_01

Pulsando el botón *Next* se llega a una segunda ventana, Figura 73, donde es posible modificar la ruta de instalación predeterminada: *"C:\Program Files\Universidad de Cadiz\PdC_01\".*

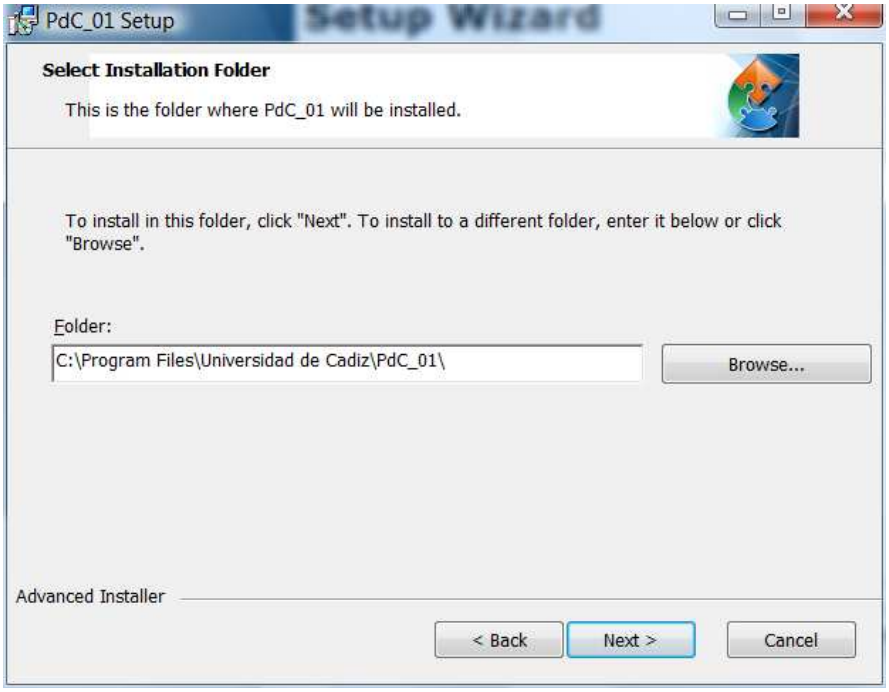


Figura 73. Selección de la ruta de instalación

Tras otras dos pantallas informando sobre el progreso de la instalación aparece una última comunicando que la aplicación ya está instalada.

Al término de la instalación se habrán creado dos accesos directos en el escritorio denominados *ResultadosPdC2* y *AsistenteFPGA_01*[33].

[33] Para este acceso directo si se utiliza Windows Vista habrá que marcar la casilla correspondiente a "Ejecutar este programa como administrador" en el Nivel de Privilegios de la pestaña Compatibilidad de sus propiedades.

Manual de usuario de la aplicación AsistenteFPGA

1. Seleccionar FPGAs

Por defecto la base de datos adjunta a la aplicación contiene los resultados ya leídos durante el trabajo experimental de la tesis, y recoge toda la información conseguida sobre la muestra de 246 modelos de FPGA sometidas al PdC propuesto. Por tanto si se desea seleccionar y clasificar FPGAs, solo será necesario ejecutar la aplicación ***AsistenteFPGA_01***.

La primera ventana informa de la espera previa a la disponibilidad del asistente, pulsando *Continuar* se inicia la lectura de las características de la muesta de FPGAs,Figura 74.

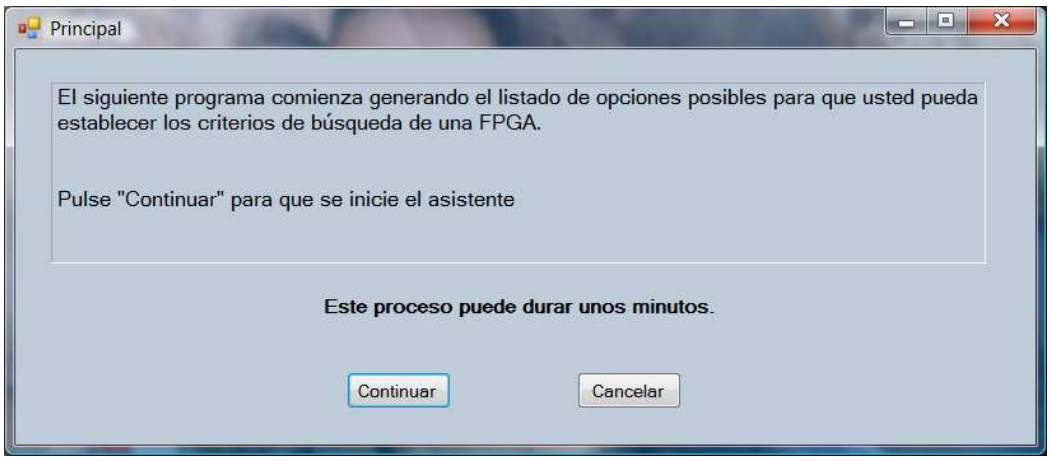


Figura 74. Ventana inicial de AsistenteFPGA

Transcurridos unos minutos esta ventana será sustituida por un cuadro de diálogo informando de que el asistente ya está listo para ser utilizado, Figura 75, desaparece al pulsar el botón *Aceptar* dando paso a la pantalla del asistente de selección de FPGAs.

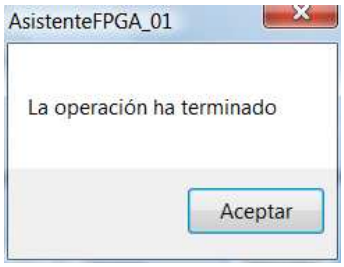


Figura 75. Cuadro de diálogo que avisa de que la búsqueda de características ha concluido

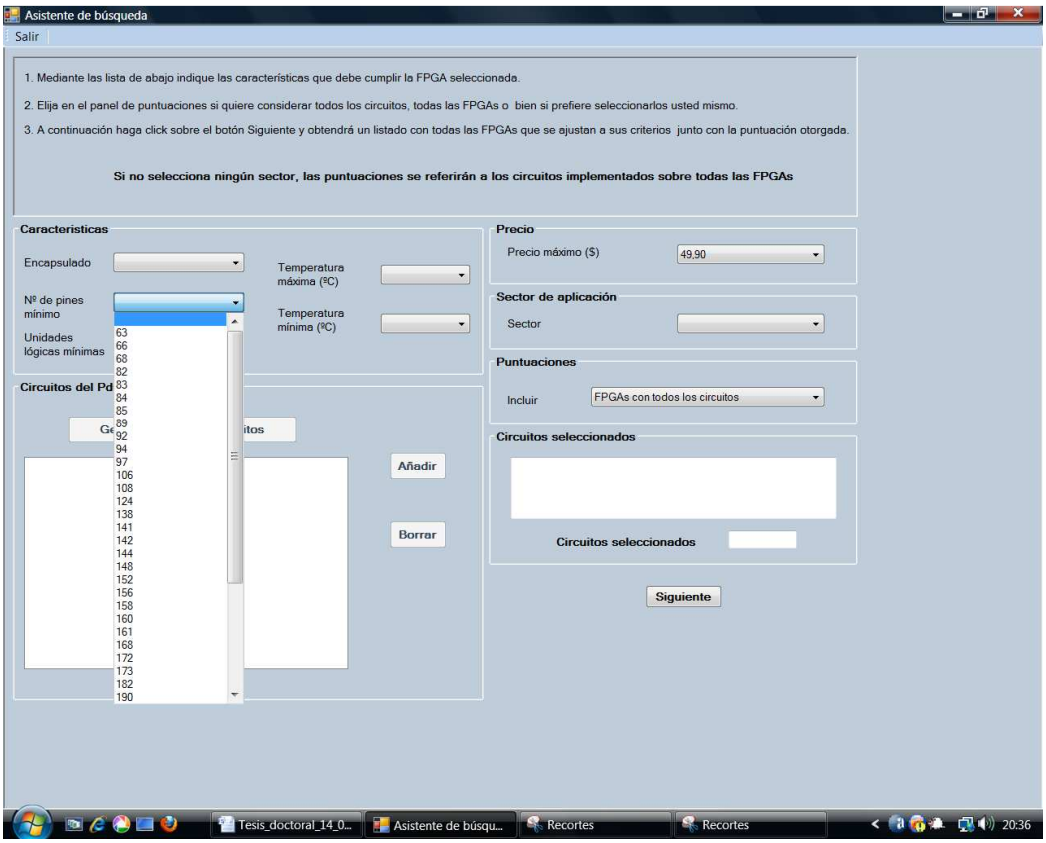


Figura 76. Pantalla principal del programa ***AsistenteFPGA_01.*** Lista "Nº de pines mínimo" desplegado

1.1. Selección de características

En el panel *Características* de la pantalla principal, Figura 77, existen cinco listas que se despliegan al pulsar con el botón izquierdo del ratón sobre ellas. Es posible seleccionar un valor en cada una volviendo a pulsar con el botón izquierdo del ratón sobre el valor deseado.

Los encapsulados aparecen con el formato de nombre utilizado por cada fabricante. Los valores seleccionados para las características *Pines, Temperatura mínima y Unidades lógicas*[34] marcarán las necesidades mínimas de estos parámetros, realizándose la búsqueda y clasificación de todos los modelos que se ajusten a esas restricciones. La *temperatura máxima* designa el límite superior al que deben ajustarse las FPGAs seleccionadas.

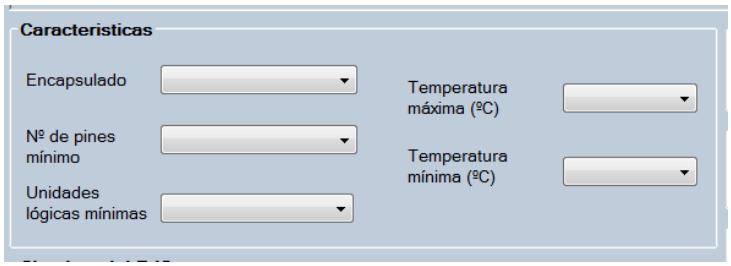


Figura 77. Panel de características

1.2. Selección de precio

El precio máximo que se desea incluir en la búsqueda y clasificación se marca mediante la única lista desplegable del panel *Precio*.

1.3. Selección de sector

La opción por defecto que presenta la aplicación corresponde a una búsqueda referida al total de circuitos que conforman el PdC, sin hacer alusión a ningún sector comercial. Es posible modificar esta condición seleccionando Automoción, Consumo o Comunicaciones utilizando el ratón sobre la lista desplegable del panel *Sector de Aplicación,* Figura 78.

[3434] Marcan el total de parejas LUT-FF

Figura 78. Panel de selección de sector

Cada sector tiene asignado un subconjunto de circuitos del PdC, asignados por defecto en el progama *ResultadoPdC2*. Es posible modificarlos mediante esta última aplicación.

Si se mantiene la opción por defecto, esto es, ningún sector, las puntuaciones otorgadas se calcularán haciendo uso de los resultados ante los 16 circuitos del PdC. Si se opta por modificar este campo y seleccionar un sector concreto, la puntuación hará referencia a los circuitos asociados previamente al sector elegido en la aplicación *ResultadosPdC*.

1.4. Conjunto de aplicación

El panel *Puntuaciones* permite establecer tres grupos de resultados a partir del filtrado establecido por el usuario en base a las características y precios de las FPGAs:

- **FPGAs con todos los circuitos (Opción por defecto):** Puntuación y clasificación de todas las FPGAs que hayan alojado todos los circuitos correspondientes a la selección del panel *Sector de Aplicación.*

- **Todas las FPGAs con circuitos comunes:** Puntuación y clasificación de todas las FPGAs analizadas utilizando solo los resultados derivados de los circuitos implementados en todas ellas.

- **Circuitos del PdC seleccionados de una lista:** Puntuación y clasificación de todas las FPGAs filtradas utilizando solo los resultados derivados de los circuitos seleccionados. Si se marca

un circuito que no haya sido implementado en todas las FPGAs, no será tenido en cuenta en la puntuación ni en la clasificación.

Si se opta por el tercer grupo, se activará el botón *Generar Listado de circuitos*, al pulsarlo mostrará un listado con el nombre de todos los circuitos que conformen el PdC en ese momento, Figura 79. A continuación, mediante el botón izquierdo del ratón será posible ir marcando todos los circuitos de interés. Finalizado este proceso de selección bastará con pulsar el botón *Añadir* para visualizar en el panel *Circuitos seleccionados* el total de elegidos y su nombre, Figura 80.

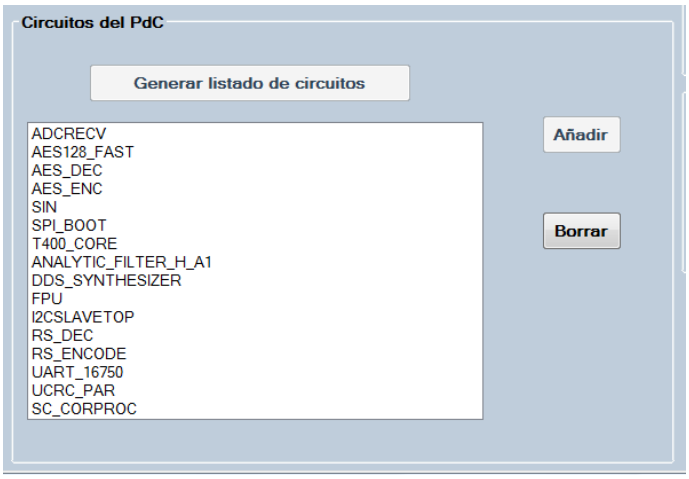


Figura 79. Panel de selección de circuitos a considerar en la puntuación y clasificación

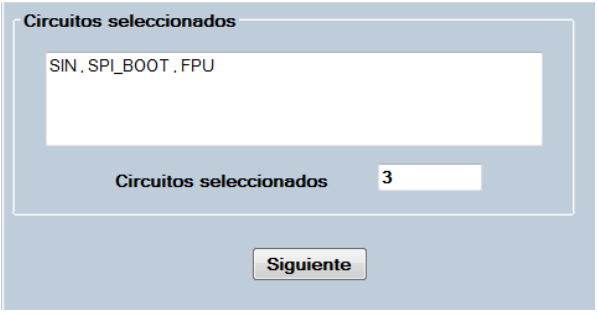


Figura 80. Panel *Circuitos seleccionados* mostrando un ejemplo de selección de circuitos

El proceso de puntuación y clasificación comenzará tras pulsar el botón *Siguiente*.

Manual de la aplicación ResultadosPdC

El objetivo de esta aplicación es automatizar la lectura de los informes generados tras la implementación del PdC sobre un conjunto de FPGAs. Finalizada la lectura la aplicación procede a formatear todas las magnitudes y realiza el cálculo y normalización de las puntuaciones. A su vez, permite seleccionar qué circuitos del conjunto total del PdC se asocian a cada uno de los tres sectores considerados: Automoción, Consumo y Comunicaciones.

Tras ejecutar esta aplicación una vez y finalizado todo el proceso de lectura y cálculo, ya no será necesario volver a utilizarla salvo que se disponga de un nuevo PdC.

Por defecto, **ResultadosPdC** ya ha sido ejecutado para obtener toda la información relativa al PdC propuesto para esta tesis. Por tanto, es posible utilizar la aplicación **AsistenteFPGA** sin abrir ResultadosPdC.

A continuación se explica su funcionamiento básico tanto para el caso de desear iniciar un nuevo proceso de lectura como para avanzar por sus ventanas y ver los distintos datos leídos.

Al iniciar el programa éste muestra una ventana, *Lectura de Informes,* con una barra de menú y tres tablas cuyo contenido se detallará más adelante. Mediante el botón de la barra de menú **Ver** se tiene acceso a otras tres ventanas. En las cuatro ventanas que integra la herramienta la información se presenta en forma de tabla. Debajo de cada tabla existe un campo que informa del total de registros que componen cada una. Las columnas de todas las tablas tienen las siguientes propiedades:

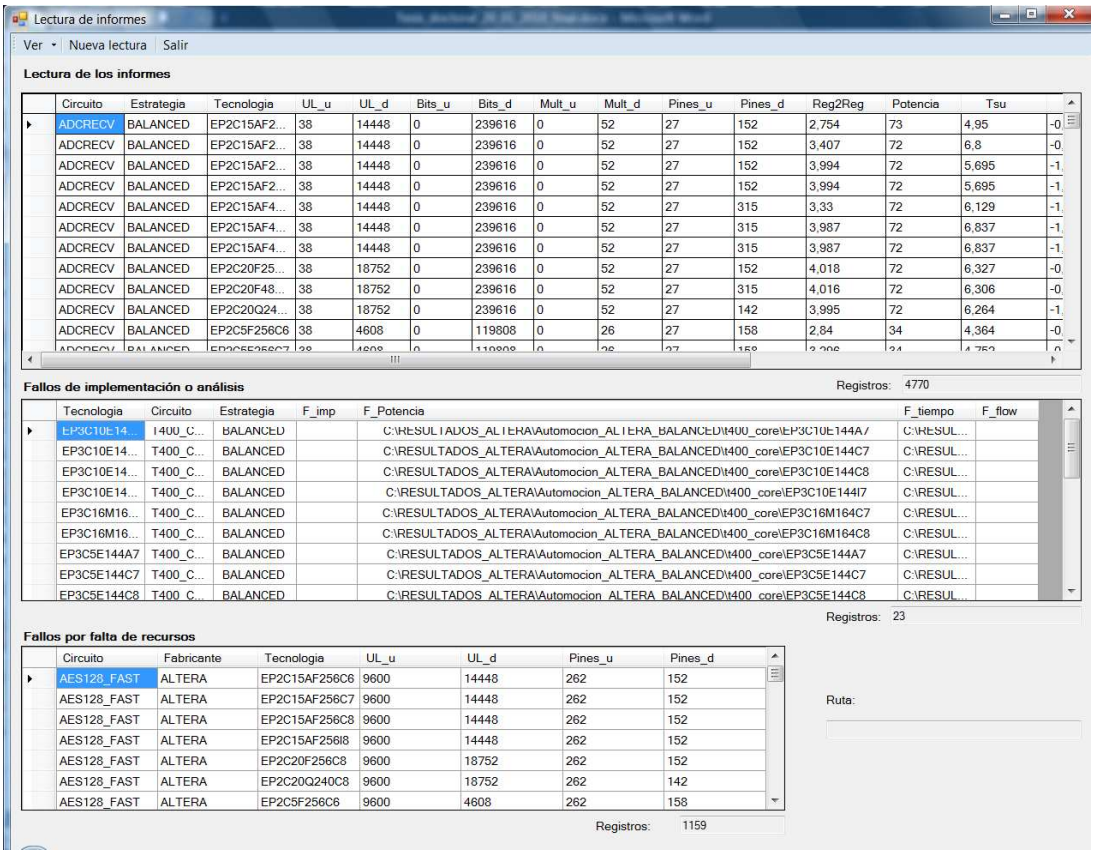


Figura 81. Ventana inicial de ***ResultadosPdC***

- **Ordenación de registros por columna**: Es posible ordenar sus registros de mayor a menor, y viceversa, con solo pulsar en el encabezado de cada una.

- **Redimensionado de columnas**: Si la tabla es demasiado grande, esto es consta de un elevado número de columnas, aparecerá una barra de desplazamiento debajo de ella que permitirá visualizarlas todas. El ancho de las columnas puede ser redimensionado colocando el cursor del ratón en las líneas divisorias y arrastrándolo hasta conseguir el tamaño adecuado para su correcta lectura. Esta propiedad es útil cuando el contenido de una celda es demasiado extenso.

En cada ventana el contenido de la barra de menú difiere con excepción del botón **Ver y Salir** siempre presentes. Con el botón **Ver** es posible ir cambiando de una ventana a otra.

1.1. Barra de menú de la ventana 1: *Lectura de Informes*

El menú de esta ventana cuenta con tres botones, Figura 82, siendo el primero de ellos desplegable, botón *Ver*.

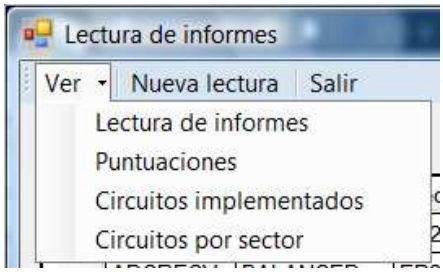


Figura 82. Detalle de la barra de menú de **ResultadosPdC**

Botón Ver

Mediante este botón se accede a las cuatro ventanas que conforman la aplicación, mostrándose en cada una distintas lecturas o resultados presentados siempre en formato de tabla. Mediante el cursor y el botón izquierdo del ratón es posible seleccionar la ventana deseada.

- **Lectura de Informes (Ventana 1)**: es la ventana principal e incluye tres tablas.
    - *Lectura de resultados.* Incluye campos relativos a todos los valores leídos en los informes de las herramientas de diseño y análisis, independientemente de que participen en el sistema de puntuaciones o no.

- *Fallos de implementación o análisis.* Lista las rutas donde deberían encontrarse los informes no hayados.
- *Fallos por falta de recursos.* Recopila información sobre los casos de FPGAs cuoys recursos son insuficientes para implementar algún circuito.

- **Puntuaciones (Ventana 2):** en esta ventana se presentan tres tablas con información referente al cálculo de puntuaciones:
    - *Cálculo de Pk.* Presenta las magnitudes de referencia asi como las cinco puntuaciones obtenidas por cada circuito y FPGA.
    - *Puntuación normalizada PK' (1 a 10).* Recoge las nuevas puntuaciones obtenidas (Pk') tras normalizar las Pk ya calculadas. Los nuevos valores estarán en el rango de 1 a 10.
    - *Fallos durante el cálculo de Pk:* Si ha habido un fallo durante el cálculo de las Pk, lo notifica en esta tabla, especificando el origen del error, el circuito y el modelo de FPGA.
- **Circuitos implementados (Ventana 3):** Mediante una única tabla se informa de qué circuitos se han implementado con éxito en cada una de las FPGAs analizadas.
- **Circuitos por sector (Ventana 4):** Presenta en tres tablas el nombre de los circuitos del PdC que se han asociado a cada sector.

Botón *Nueva Lectura*

Este botón inicia una nueva lectura de resultados tras haber concluido la ejecución de un nuevo PdC. Al pulsar con el ratón

sobre él se abre una ventana de búsqueda dónde debe indicarse la ruta de almacenamiento de los resultados, Figura 83.

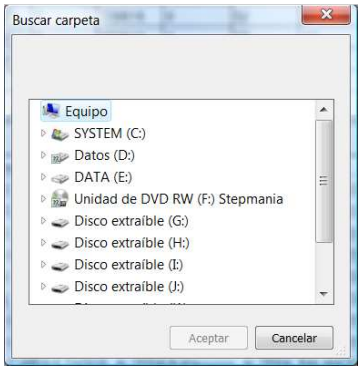


Figura 83. Ventana de búsqueda

Tras seleccionar un camino de búsqueda la herramienta iniciará la lectura de todos los informes y a su término mostrará los resultados en las tablas de la ventana principal así como la ruta de búsqueda utilizada. Este proceso es muy lento y dependiendo del número de ficheros a leer requerirá de más o menos tiempo.

Botón Salir

Cierra la aplicación.

1.2. Barra de menú de la ventana 2: *Puntuaciones*

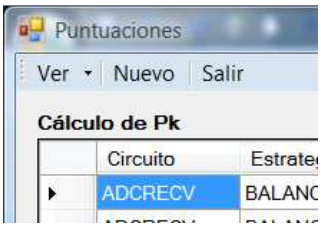


Figura 84. Barra de menú de la ventana *Puntuaciones*

Botón Ver

Su funcionamiento y contenido coincide con el de la ventana *Lectura de Resultados.*

Botón Nuevo

Calcula las nuevas puntuaciones Pk y Pk' tras haber iniciado una fase de lectura de resultados en la ventana anterior.

Botón Salir

Cierra la aplicación.

1.3. Barra de menú de la ventana 3: *Circuitos Implementados*

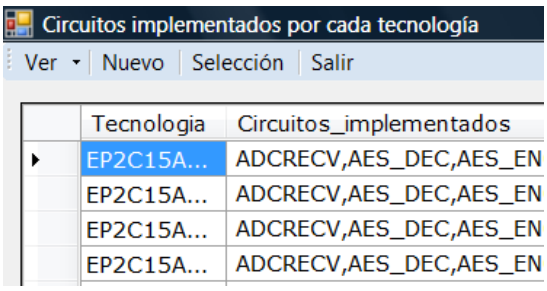


Figura 85. Barra de menú de la ventana *Circuitos Implementados*

Botón Ver

Su funcionamiento y contenido coincide con el de la ventana *Lectura de Resultados.*

Botón Nuevo

Obtiene la nueva lista de circuitos implementados con éxito en cada FPGA, tras haber iniciado una nueva lectura de resultados en la ventana inicial.

Botón Selección

Abre una nueva ventana, **Selección de circuitos por sector,** que permite seleccionar los circuitos que se desea asociar a cada sector de mercado.

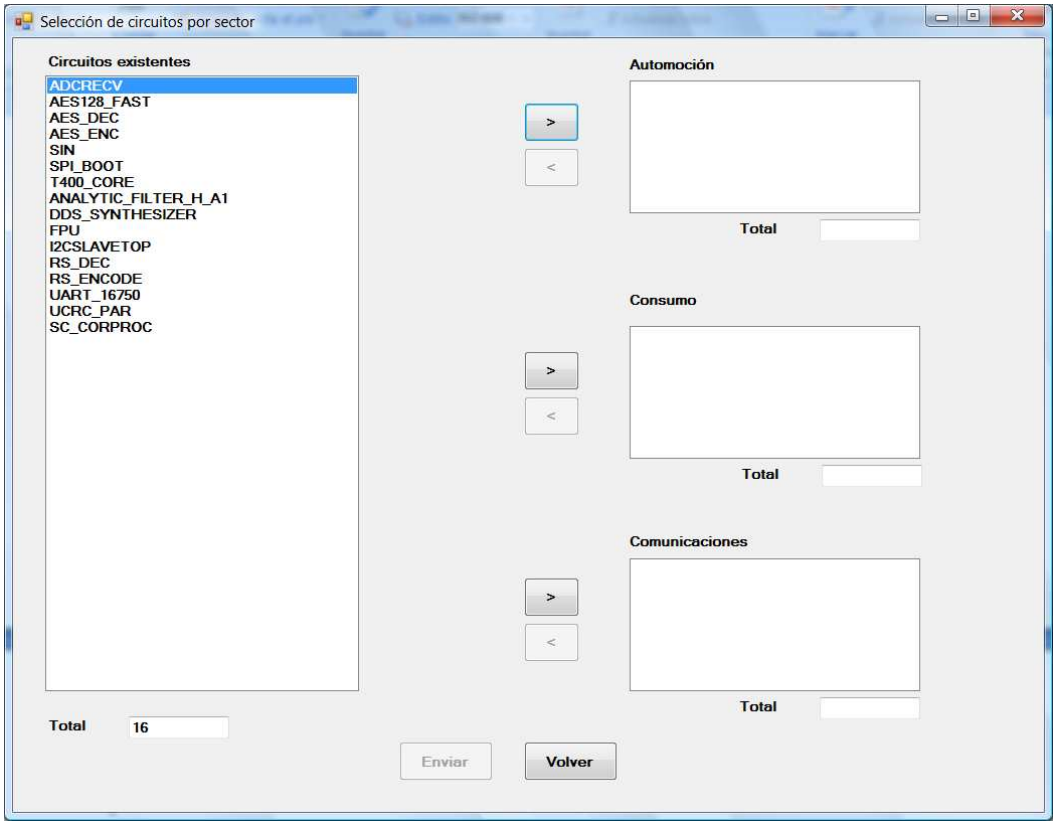


Figura 86. Ventana de selección de circuitos por sector

Con el ratón deben seleccionarse todos los circuitos a agregar a un sector antes de pulsar sobre el botón >.Concluida la tarea de asociar circuitos a todos los sectores deben enviarse a la base de datos mediante el botón **Enviar.** Pulsar **Volver** para ir a la pantalla **Circuitos Seleccionados por sector** (Ventana 4) donde estarán.las tablas de los circuitos asociados a cada sector.

Botón Salir

Cierra la aplicación.

1.4. Barra de menú de la ventana 4: *Circuitos seleccionados por sector*

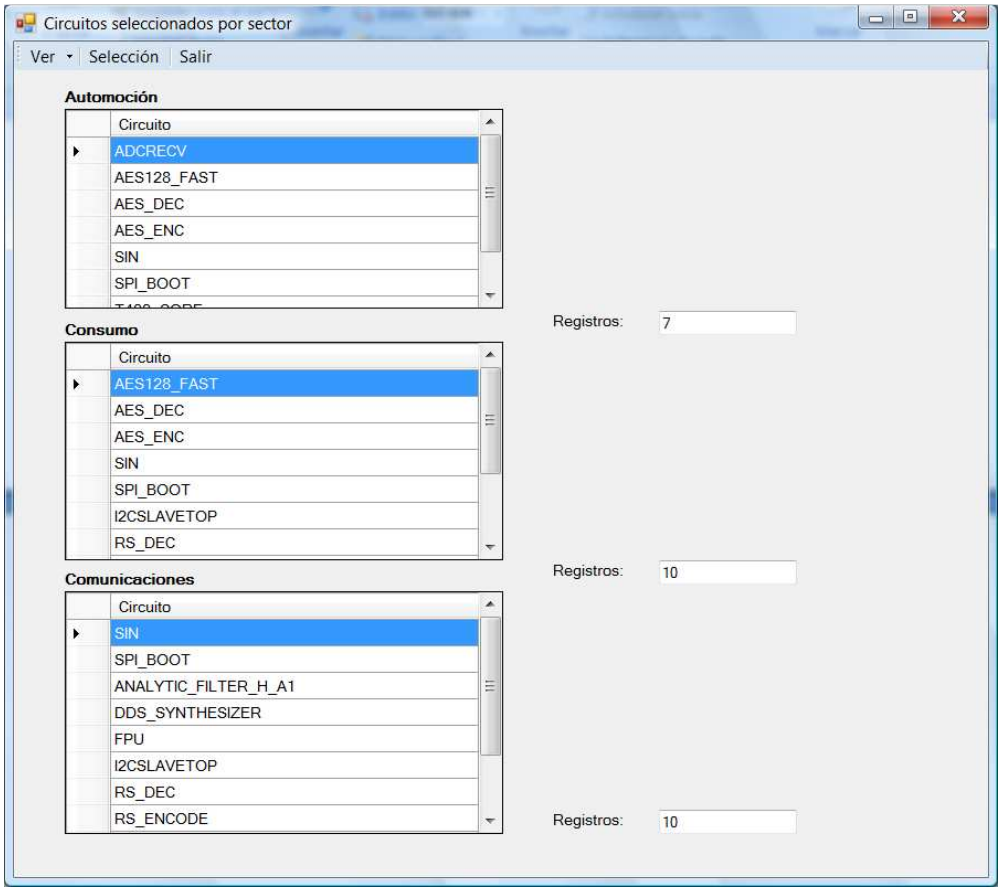


Figura 87. Barra de menú de la ventana **Circuitos Seleccionador por sector**

Botón Ver

Su funcionamiento y contenido coincide con el de la ventana *Lectura de Resultados.*

Botón Selección

Abre la ventana de **Selección de circuitos por sector** explicada en el apartado anterior

Botón Salir

Cierra la aplicación.